NPS69-86-005

NAVAL POSTGRADUATE SCHOOL

Monterey, California

CONTRACTOR REPORT

THRUST VECTOR CONTROL,

HEAT TRANSFER MODELING

by

A. Leitner

July, 1986

Approved for public release; distribution unlimited.

Prepared for:
Naval Postgraduate School
Monterey, California 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

RADM R. C. Austin
Superintendent

D. A. Schrady
Provost

The work reported herein was carried out for the Naval Postgraduate School by Mr. Amiram Leitner under Contract Number N62271-86-M-0206. The work presented in this report is part of a project sponsored by Naval Weapons Center on heat transfer modeling and is under the cognizance of Professors Matthew D. Kelleher and Robert H. Nunn.

This report was prepared by:

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S): NPS69-86-005	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION: Naval Postgraduate School; 6b OFFICE SYMBOL (If applicable)	7a NAME OF MONITORING ORGANIZATION: Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000	7b ADDRESS (City, State, and ZIP Code): Monterey, California 93943-5000
8a NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Weapons Center; 8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N62271-86-M-0206
8c ADDRESS (City, State, and ZIP Code): China Lake, California 93555	10 SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO; PROJECT NO; TASK NO; WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)
Thrust Vector Control, Heat Transfer Modeling

12 PERSONAL AUTHOR(S)
A. Leitner

13a TYPE OF REPORT	13b TIME COVERED FROM ____ TO ____	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT 65

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.

Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.

The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code); 22c OFFICE SYMBOL

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Thrust Vector Control Heat Transfer Modeling

Abstract

The report presents heat transfer modeling of Thrust Vector control systems using the PHOENICS computer code.

Simple two-dimensional wedge and blunt bodies have been examined in supersonic cold flow, for both laminar and turbulent flow cases.

The research presents a numerical solution of the supersonic compressible viscous two-dimensional flow field. Post calculations were done to estimate skin friction coefficient, surface heat flux, heat transfer coefficient and Stanton number distributions in both wedge and blunt cases.

NOMENCLATURE

C_p	Specific heat [J/kg•k]
C_1, C_2, C_D	Constants used in turbulent model
C_f	Skin friction coefficient
h	Enthalply [J/kg]
h_c	Heat Transfer coefficient [W/m^2k]
M	Mach number
P	Pressure
Pr	Prandtl number
q	Heat flux
R	Gas constant [J/kg•k]
Re	Reynolds number
St	Stanton number
t	Time [S]
T	Temperature

GREEK LETTER SYMBOLS

γ	Specific heat ratio
δ	Boundary layer thickness
μ	Dynamic viscosity [kg*m/s]
σ	General exchange coefficient
ρ	Density [kg/m^3]
τ_k, τ_ε	Constants used in turbulent model
ϕ	Any property at the grid node

SUBSCRIPTS

comp	Compressible value
eff	Effective value
inc	Incompressible value

r	Recovery
lam	Laminar quantity
t	Turbulent quantity
stat	Static values
W	Wall value
z	Local value in the flow direction
∞	Free stream value

Table of Contents

List of Figures

1. Introduction

This report describes a numerical analysis of heat transfer of a typical jet vane configuration used for thrust vector control. The work was carried out under contract Nos. N62271-85-M-0443 and N62271-86-M-0206, for the Naval Postgraduate School.

The tasks to be accomplished under the first contract were:

Task I: Formulate the conservation equations of momentum energy for two-dimensional, supersonic flow in geometries typical of thrust vector control systems.

Task II: Formulate boundary conditions for these equations appropriate to thrust vector control systems.

The tasks to be accomplished under the second contract were:

Task I: Continue and update the formulation of thrust vector control geometries based on the input from the Naval Weapons Center (NWC).

Task II: Construct the computational model for implementation in the PHOENICS code, of the thrust vector control geometries and flow conditions provided by NWC.

Task III: Run the PHOENICS code for the previously formulated models. Analyze and interpret the PHOENICS results for surface temperature and heat flux.

Thrust vector control components such as jet vanes and jet tabs are exposed to high speed hot gases at the exit of a rocket nozzle.

Estimation of the heat transfer from the hot exhaust gases to the vane is major consideration in the correct design of a vane, and its ability to survive during its mission.

The research work was done under the framework of the tasks. A brief survey of what has been done according to the task is given:

Task 1 (M-0443): Heat transfer modeling of thrust vector control vane requires supersonic compressible viscous flow analysis.

In order to meet the requirements, the conservation differential equations of mass momentum energy and the two $k-\varepsilon$ turbulent equations were formulated, and additional algebraic formulas for the relations between pressure density and the equation of state for ideal gas.

Task II (M-0443): The physical dimensions of the flow field grid were chosen and the boundary conditions for the Navier-Stokes, energy and the two $k-q$ turbulent model equations were given.

Task I (M-0206): Working on the task, the actual configuration of a jet vane that is presently being tested at NWC has been modeled. The geometry being used is a wedge which has the same half angle and dimensions as the NWC jet vane.

Task II (M-0206): BFC (Body fitted coordinate) version of PHOENICS code (Ref. 3) was used for calculating the flow-field and heat transfer over the model. Using the BFC, a better geometrical approximation to vane shape could be achieved.

Non-Uniform grids have been utilized in order to model complicated regions in the flow field. Relaxation parameters and false timesteps options were adjusted to enable efficient computer runs with good convergence.

Task III (M-0206): In carrying out this task, four major runs have been analyzed:

Two geometric configurations were used: wedge vane and blunt vane (see Figures 1, 2, 3, 4); each one in both laminar and turbulent flow conditions.

Numerical results for fluid field and thermodynamic properties of pressure, temperature, density, Mach number and velocities are given in appendix C.

Post-calculations of heat transfer coefficient, skin friction coefficient and Stanton number are given in Figures (6, 7, 8, 9, 10, 11).

The next chapters describe in more detail the process of building the model and the analysis of the results.

2. PHOENICS Description

The present work addresses the heat transfer modeling of thrust vector control systems. In this effort the Navier-Stokes approach is applied by using a computer code which is capable of simulating a large number of fluid flow, heat transfer and chemical reaction processes which arise in industry and elsewhere. This code is called PHOENICS, which is an acronym standing for: 'Parabolic, Hyperbolic or Elliptic Numerical Integration Code Series.' The name comes from the fact that the differential equations of fluid flow, etc. arise in forms classified by mathematicians as parabolic, hyperbolic or elliptic; and PHOENICS solves these equations, whatever their form.

Built into PHOENICS are the major conservation laws of physics (mass, momentum, and energy) applied to a large number of continuous subdomains called 'cells,' into which the domain of study is artificially divided. The number of cells can be few or many according to the requirements of the problem. Because of numerical stability considerations the restrictions on cell refinement can become particularly burdensome in the calculation of a turbulent boundary layer where a very fine mesh near the wall may be required.

When supplied with appropriate information concerning: the physical properties of the materials, the geometrical and other constraints, the inlet and/or initial conditions, PHOENICS computes the corresponding solutions to the relevant differential equations, expressing them as tables of numbers describing the field of velocity, temperature concentration, etc.

Detailed information about PHOENICS is given in [Ref. 3].

2.1 The Structural Principle of PHOENICS

The code consists of three major parts: Satellite subroutine, Ground subroutine and Earth library.

The satellite subroutine is the main input subroutine and should provide the answers to the questions:

- what kind of process is to be simulated
- what are the properties of the fluid
- what are the shape and size of the domain
- how fine is the grid to be employed
- to what degree of accuracy is the calculation to be continued
- and what output should be provided

Ground subroutine is active during the computing process and is used for updating properties which vary with time, temperature, etc. For example: viscosity depends on temperature or density depends upon pressure and temperatures, etc.

Earth library is the main solver generator. It is given as a binary library and does not enable the user access to the source code.

2.2 Numerical Scheme

The numerical scheme used by the code is the simpler (semi-implicit method for pressure-linked equations revised) (Ref. 9). The scheme was developed by Patankar, S. V. and Spalding, D. B.

The scheme requires an additional dependent variable, the pressure correction, which has no physical meaning but should take part in the process.

The value of the pressure correction should tend to zero in the convergence process.

Two additional differential equations are solved: for the pressure, and for the pressure correction.

3. Geometry and Dimensions

Symmetrical 2-D planar geometry, which is shown in Figure 2, was chosen to be the approximation of the MWC vane in Figure 1.

Two geometrical profiles were examined, one with wedge leading edge and the second with blunt leading edge.

The dimensions of the domain in Figure 3 and 4 satisfy aspect ratio of 10:1 in the vertical y coordinate. A high aspect ratio in the coordinate is important for the assumption of free stream conditions at the upper boundary.

4. Assumptions

Postulating the right or the wrong assumptions has the most influence on modeling process. The stage was carried out very carefully in order to make the most compatible model with reality.

4.1 Steady state:

The modeling assumes steady state physical phenomenon process.

$$\frac{\partial}{\partial t} (\text{all properties}) = 0$$

This is a valid assumption since the time constant for the convection process is much shorter than the time constant for the wall conduction.

By assuming the wall temperature to be constant, the two procedures are decoupled.

In hot flow it is important to run the code for a wide range of wall temperature which will take into account the influence of different temperatures on the heat convection process.

4.2 Cold Air Flow

Ambient temperature air flow which was utilized by NWC experiments is being used in the computations.

4.3 Ideal Gas

The gas is assumed to satisfy the ideal gas equation of state

$$p = \rho RT \tag{4.1}$$

This is a fairly good assumption for nonreactive gas flow. In spite of the values of static temperature can decrease to 200[k], the density remain relatively low.

This assumption is an important simplification to the solution in Ref. 10 which used the isentropic relation between pressure and density instead

$$\frac{\rho}{\rho_o} = \left(\frac{P}{P_o}\right)^{1/\gamma} \tag{4.2}$$

4.4 Constant Pr,γ:

Prantdl number and γ (ratio of specific heats) were found to have negligible variations in the temperature range of the model. (200k ÷ 350k)

4.5 Varying Viscosity and Thermal Conductivity:

μ and k are much more dependent on temperature especially very close to the solid wall where values of μ and k influence strongly the shear and heat transfer mechanism. To account for the temperature dependence power law relations have been formulated for μ and k.

$$\mu = \mu_o\left(\frac{T}{T_o}\right)^{0.666} \tag{4.3}$$

$$k = k_o\left(\frac{T}{T_o}\right)^{0.666} \tag{4.4}$$

4.6 Parallel Flow

Gas flow at the exit of the exhaust nozzle is more likely to be a conic source flow than parallel flow.

If the half angle of the nozzle is small, ($\alpha < 15°$), parallel flow is a good assumption

4.7 Negligible Radiation

Assessments that were done showed that heat convection is at least one order of magnitude greater than heat flux by radiation.

4.8 Laminar and Turbulent Solutions

In order to overcome lack of ability to predict transition, separated laminar and turbulent calculations were done for each case. The turbulent solution utilizes the ($k-\varepsilon$) eddy viscosity model Ref. 5.

4.9 Constant Wall Temperature

The vane wall is assumed to have constant temperature during the time of calculation.

5. Governing Equations

The conservation equations for the compressible flow of the mathematical model consists of a viscous, Newtonian perfect gas consisting of the following six differential equations:

Conservation of Mass:

$$\frac{\partial}{\partial t}(\rho) + \nabla\cdot(\rho\vec{V}) = 0 \tag{5.1}$$

Conservation of momentum:

$$\frac{\partial}{\partial t}(\rho\,\phi) + \nabla\cdot\,(\rho\vec{V}\phi - \mu\nabla\phi)\ \nabla P \tag{5.2}$$

where ϕ is V or W velocity component for y and z direction.

Conservation of Energy

$$\frac{\partial}{\partial t}\ (\rho h) + \nabla\cdot(\rho\vec{V}h - \frac{\mu}{P_r}\ \nabla h) = \frac{Dp}{Dt} \tag{5.3}$$

where h is the total enthalpy.

$$h = C_p\ T_o$$

where To is the total temperature

$$T_o = T_{stat}*(1 + \frac{\gamma-1}{2}M^2)$$

In the case of laminar flow the governing equations (5.1), (5.2), (5.3) are sufficient to determine a solution when proper boundary conditions are applied and the equation of state (4.1) is provided.

Turbulence Model:

In turbulent flow it is necessary to hypothesize a turbulence ɔdel relating the turbulent viscosity to the other problem variables.

The model used in PHOENICS is the eddy viscosity (k-ε) model [Ref. 3, Ref. 5]. In this model k, the turbulent kinetic energy and ε, the turbulence dissipation rate, are treated as properties of the flow and conservation equations are postulated for these properties. The two conservation equations are: one for k, the kinetic energy of turbulence:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial Xj}\left(\frac{\nu_{eff}}{\sigma_k}\frac{\partial k}{\partial X_j}\right) + G_k - \varepsilon \tag{5.4}$$

Second equation for ε, the dissipation rate of turbulence

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial Xj}\left(\frac{\nu_{eff}}{\sigma_\varepsilon}\frac{\partial \varepsilon}{\partial X_j}\right) + \frac{\varepsilon}{K}\left(C_1 G_k - C_2\varepsilon\right) \tag{5.5}$$

where

$$G_k = \nu_t \left(\frac{\partial \overline{U}i}{\partial X_j} + \frac{\partial \overline{U}j}{\partial Xi}\right)\frac{\partial \overline{U}i}{\partial Xj} \tag{5.6}$$

$$\mu_{eff} = \mu_{lam} + \rho c_\mu k^2/\varepsilon \tag{5.7}$$

c_1, c_2, σ_k, σ_ε, c_μ are empirical constants which are provided in PHOENICS.

The reason for using the (k-ε) model is because it is the most verified model for engineering applications. It combines simplicity, universality, and realism of predictions in most cases.

Two additional differential equations are solved also in order to satisfy the SIMPLER algorithm as was mentioned in chapter 2.2. The description of the pressure and pressure correction equations is provided by Ref. 9.

6. Input Variables

The properties of mach no. stagnation presence and temperature of the gas were provided by NWC; additional properties were taken from air tables:

Mach number:

$M_\infty = 3.2$

Stagnation pressure:

$P_\circ = 55.10^5$ [Pa]

Stagnation temperature:

$T_\circ = 555.55$ [K]

Gas constant

$R = 287.$ [J/kg·k]

Specific heat ratio

$\gamma = 1.35$

Laminar Prandtl Number

$Pr = 0.7$

Turbulent Prandtl Number

$Pr_t = 0.9$

Constant Pressure Specific Heat

$C_p = R/(1-1/\gamma)$ [J/kg·k]

Laminar Viscosity

$\mu = 0.1716 * 10^{-5} * (T/273.)^{0.666}$

Thermal Conductivity

$k = \mu\ C_p/P_r$

The gas properties in the inlet boundary are equivalent to the properties at nozzle exit. Inlet properties are calculated from the stagnation values in the combustion chamber. The calculation was done by assuming one dimensional

isentropic expansion from combustion chamber to the nozzle exit (inlet for the vane).

Pressure $$P_i = P_o/(1 + \frac{\gamma-1}{2} M^2)^{\gamma/\gamma-1} \tag{6.1}$$

Temperature $$T_i = T_o/(1 + \frac{\gamma-1}{2} M^2) \tag{6.2}$$

Density $$\rho_i = P/RT \tag{6.3}$$

Enthalpy $$h_i = C_p T_i \tag{6.4}$$

Sonic Velocity $$C_i = \sqrt{\gamma R T_i} \tag{6.5}$$

Velocity $$W_i = C \cdot M \tag{6.6}$$

The subscript i signifies inlet property

7. Boundary Conditions

The flow field described in Figures 3, 4 has four boundaries, which can be named: inlet, outlet, freestream boundary and solid wall.

Super sonic flows have a hyperbolic mathematical nature. The field consists of influence zones, the flow at every point is governed only by its influence zone, basically by the upwind stream.

As a consequence from the discussion, it's obvious that the outlet boundary condition has no influence on the upstream flow. The boundary values that are given at the outlet are to satisfy some numerical needs only.

7.1 Inlet

Parallel uniform flow with known velocity, enthalpy, pressure, and density: equation (6.1), (6.3), (6.4), (6.6) are given at the left boundary of the grid. In PHOENICS this is specified as the LOW side of the first Z cell.

In turbulent flow, boundary conditions are supplied for k and ε. The values that are given are based on empirical values:

$$k_i = 0.0\ W_i^2 \tag{7.1}$$

$$\varepsilon_i = 0.16\ k^{1.5}/(5*GH) \tag{7.2}$$

where GH is half the vane thickness

7.2 Outlet

As was mentioned previously, the outlet has negligible effect on the results. The only property that is specified at the outlet is the pressure.

7.3 Freestream Boundary

Assuming that the upper boundary is chosen to be far enough away, the default boundary condition option of PHOENICS is used. This implies a line of symmetry where all gradients are zero.

7.4 Solid Wall

Zero velocity and constant wall enthalpy (temperature) are assumed on the wall. In PHOENICS the wall is the SOUTH side of the first y cell. The high enthalpy and velocity gradients near the wall demands a refined grid close to the wall. Values of shear stress and heat flux are calculated to first order accuracy using:

$$\tau_w = \mu \frac{\partial W}{\partial y} \simeq \mu \frac{W_1}{\Delta y_1/2} \tag{7.1}$$

$$q_w = \frac{\mu}{Pr} \frac{\partial h}{\partial y} \simeq \frac{\mu}{Pr} \cdot \frac{h_1 - h_w}{\Delta y_1/2} \tag{7.2}$$

In turbulent flow, a wall function is used to provide the wall condition for velocity, enthalpy, k, and ε

7.5 Wall Function

The wall problem in the numerical computation of flows, especially in turbulent flow, is an old one and most authors have adopted similar techniques. In effect they "bridge over" the region very close to the wall by introducing special functions which are called wall functions. These are often empirical in origin. Accounts may be found in Ref. 11.

The problem arises as follows. Turbulence dies out, close to the wall, because the no slip condition and the rigidity of the wall make all the velocity components fall to zero. The consequence is that the effective

viscosity and other transport properties fall there to their laminar values and the result is a rapid variation with distance from the wall both of the ϕ 's and of their gradients.

Where ϕ signifies general dependent variable, it is possible to compute these variations in detail, by using a computer code such as PHOENICS on two conditions:

(i) the grid points must be packed into the region of steep gradient changes closely enough for sufficient numerical accuracy to be obtained

(ii) the functions appearing in the turbulence model equations must properly represent the influence of local Reynolds number on turbulence.

Under the conditions above, the wall function sequences in the program act as follows:

The Reynolds number is first evaluated, based on the resultant velocity parallel to the wall, on the distance from the wall to the grid node and on density and laminar viscosity. If this Reynolds number is less than 132.25 (the value at which the laminar and turbulent wall function intersect) a laminar wall function is used. If this Reynolds number turns out to be greater than 132.25 the velocity variation is logarithmic and the corresponding shear stress coefficient is evaluated. This corresponds to the commonly used "log law" wall function. [Ref. 4]

7.6 Boundary Conditions in Phoenics

PHOENICS utilizes source terms for creating boundary conditions. The form of the source term of each dependent variable ϕ is:

$$S\phi = ([\tilde{m}] + C\phi)\ (V\phi - \phi_p) \tag{7.3}$$

where: m - is mass flux source

ϕ_p - is the value of the dependent variable at point near the boundary

$C\phi$, $V\phi$ - two coefficients specified by the user. The source term for

mass flux is simply

$$S_m = C_m (V_m - P_p) \tag{7.4}$$

where: Pp -is the pressure near the boundary and C_m, V_m are two coefficients.

The values of $C\phi$ and $V\phi$ for the dependent variables in SATELLITE are: At the Inlet:

$$C_m = 2 \frac{\gamma}{\gamma-1} \frac{1}{Wi} \tag{7.5}$$

$$V_m = P_o \, \rho_i / P_o \tag{7.6}$$

$$C_w = C_h = C_k = C_\varepsilon = 0. \tag{7.7}$$

$$V_w = W_i \tag{7.8}$$

$$V_h = h_i \tag{7.9}$$

$$V_k = K_i \tag{7.10}$$

$$V_\varepsilon = \varepsilon_i \tag{7.11}$$

At the Outlet:

$$C_m = 1000 * W_i \cdot \rho_i / P_i \tag{7.12}$$

$$V_m = P_i \tag{7.13}$$

At the Wall (laminar)

$$C_w = \mu/(0.5 \Delta\mu_1) \tag{7.14}$$

$$V_w = o \tag{7.15}$$

$$C_h = \mu/Pr/(0.5 \cdot \Delta\mu_1) \tag{7.16}$$

$$V_h = C_p * T_w \tag{7.17}$$

At the Wall (turbulent)

$$C_w = C_h = C_k = C_\varepsilon = WALL \tag{7.18}$$

$$V_w = V_k = V_\varepsilon = 0 \tag{7.19}$$

$$V_h = Cp*Tw \tag{7.20}$$

8. Mesh Generation

In this work a two-dimensional mesh is being used with 18 x 29 cells in the y and z coordinate respectively. A Nonuniform grid has been used for both directions. Figures 3 and 4 shows the grid in the z direction. A finer grid is used in the blunt region, IZ = (7 ÷ 17), and in the zone, where the inclined wall transitions to a straight wall, IZ = (23 ÷ 26).

In the y coordinate, except in the boundary layer region, the grid is uniform. To obtain a finer grid resolution in the boundary layer for the laminar flow case the first five cells in the y direction from the wall obey the following proportionality relationship:

$$BYFRAC\ (IY) = \left(\frac{IY}{5}\right)^3 \left(\frac{\Delta_{max}}{10GH}\right) \quad (8.1)$$

Where BYFRAC(IY) is the distance from the south side to the north side of the cell of particular interest, divided by total length of the domain, IY is the cell number, Δ_{max} is maximum allowable cell height, and GH is the half thickness of the TVC jet vane.

A fine grid resolution for the turbulent flow case is set up in the same way as laminar flow. The only difference comes from the selection of the first five cells in y direction. The following calculation shows the difference.

From the laminar solution and the given properties the following are known:

$w = 885.2[m/s]$

$\mu_{lam} = 1*10^{-5}\ [N.s/m]$

$Po = 5.5*10^6[Pa]$

$Pstatic = 1.048*10^5[Pa]$

$\gamma = 1.35$

$\rho = 1.835$ [kg/m]

Using the values above and the length of vane, which is 0.095m, A corresponding Reynolds number was calculated:

$$Re_z = \frac{\rho_\infty W_\infty Z}{\mu_{lam}} = \frac{(1.835*888.5*0.095)}{1*10^{-5}} = 1.54 * 10^6$$

Using a power law correlation for the boundary layer thickness:

$$\frac{\delta}{z} = 0.37 * Re_z^{-1/5} \qquad (8.2)$$

From equation (8.2) the boundary layer thickness at the high end of the domain has been calculated as $\delta \approx 2 * 10^{-3}$[m]

With Re based on W_∞ the velocity parallel to the wall, $\frac{\Delta y}{2}$ the distance from the wall to the first grid node, ρ_∞ the density, and μ_{lam} the laminar viscosity, Δ_y must satisfy the condition

$$R_{e\Delta} = \frac{\rho_\infty W_\infty \Delta_Y}{2 \mu_{lam}} > 132.25 \qquad \text{or} \qquad \Delta Y > 6.48 * 10^{-6}[m]$$

Therefore the interval of Δy is chosen such that

$$2 * 10^{-3}[m] > \Delta Y > 6.48 * 10^{-6}[m]$$

In this effort using the relationship

$$BYFRAC(IY) = \left(\frac{IY}{5}\right)^2 \left(\frac{\Delta_{max}}{10GH}\right)$$

Δy has been calculated as $\Delta y = 8 * 10^{-5}$[m] which is in the required interval.

For both the laminar and turbulent cases, cells in the z direction were adjusted so that the points where possible physical phenomena such as shock waves and expansion fans are expected, very fine cells were used. In the other parts of the domain larger cells were used.

9. Heat Transfer Analysis

Skin friction and heat transfer quantities were calculated in both laminar and turbulent cases and they are shown in Figures (6 ÷ 11).

9.1 Laminar Calculation

In laminar flow fluxes can be derived directly from the gradients near the wall. The first cell is close "enough" to the wall and gradients of velocity and enthalpy do not change much in this region near the wall. The shear stress and heat flux in the laminar case will be:

$$\tau_w \simeq \mu \frac{W_1}{\Delta Y_1/2} \tag{7.1}$$

$$q_w \simeq \frac{\mu}{P_r} \frac{h_1 - h_w}{\Delta Y_1/2} \tag{7.2}$$

The skin friction coefficient and Stanton number will be:

$$C_f = \frac{2 * \tau w}{\rho_\infty W_\infty^2} \tag{9.1}$$

$$S_t = q_w/[\rho_\infty \mu_\infty (h_r - h_w)] \tag{9.2}$$

where h_r is the recovery enthalpy

$$\frac{h_r}{h_o} = \frac{1 + \frac{r(\gamma-1)}{2} M_\infty^2}{1 + \frac{(\gamma-1)}{2} M_\infty^2} \tag{9.3}$$

r- is the recovery factor

$$r = \sqrt{Pr} \text{ (laminar flow)} \tag{9.4}$$

The coefficient of heat transfer in convection was calculated using

$$h_c = \rho_\infty U_\infty C_p S_t \tag{9.5}$$

9.2 Turbulent Calculations

In turbulent flow the gradients of velocity and enthalpy near the wall are very steep and change rapidly with distance from the wall.

Direct calculation of flux gradients is not accurate in this case. The log law approach is used to calculate skin friction. In the calculations using PHOENICS flow field, the following relation has been used.

$$C_f = \frac{2 \rho_w k_w}{w_\infty^2 \rho_\infty 3.33} \tag{9.6}$$

To obtain equation 9.6, the turbulent kinetic energy equation has been used as a starting point. [Ref. 5],

$$\rho \frac{Dk}{Dt} = \frac{\partial}{\partial y} \left(\frac{\mu_t}{\delta_k} \frac{\partial k}{\partial y}\right) \tag{9.7}$$

$$+ k \left[\frac{\mu_t}{k} \left(\frac{\partial u^2}{\partial y}\right) - C_D \frac{\rho^2 k}{\mu_t} \right]$$

The source term of the turbulent kinetic energy equation should be zero near the wall which means

$$\frac{\mu_t}{k} \left(\frac{\partial u}{\partial y} \right)_w^2 - C_D \frac{\rho^2 k}{\mu_t} = 0 \tag{9.8}$$

therefore the shear stress on the wall can be defined as:

$$\tau_w = C_D^{1/2} \rho_w k_w \tag{9.9}$$

where k_w is the turbulent kinetic energy on the wall, ρ_w is the density on the wall and $C_D = 0.09$ [Ref. 5], substituting the values above into the Blasius skin friction relation the C_f equation becomes:

$$C_f = \frac{2\ \tau_w}{\rho_\infty\ W_\infty^2} = \frac{\rho_w}{\rho_\infty}\ \frac{2}{w_\infty^2}\ \frac{k_w}{3.33} \tag{9.10}$$

The heat transfer quantities are evaluated from the Chilton-Colburn form of Reynolds analogy.

$$s_t = (C_f/2) * P_r^{-2/3} \tag{9.11}$$

$$q_w = s_t * \rho_\infty * U_\infty * (h_r - h_w) \tag{9.12}$$

where equation (9.3) is used to evaluate h_r with the recovery factor given as:

$$r = P_r^{1/3} \text{ (turbulent flow)} \tag{9.13}$$

The convective heat transfer coefficient is calculated by using equation (9.5)

10. Code and Computer

PHOENICS 81, Body Fitted Coordinate (BFC) version has been used in the computations (see Ref. 3). PHOENICS has been installed on NPS IBM 3033 MVS 1.3 computer. 400 sweeps per computer run provided a reasonable convergence in all runs except the turbulent blunt case continuity error of less than 4.10^{-4} has been achieved in the three runs.

The continuity error is the total summation of the absolute mass imbalance in all cells divided by the inlet mass flux. CPU time consumption varies from case to case as follows:

Laminar Wedge	630	CPU Seconds
Turbulent Wedge	630	CPU Seconds
Laminar Blunt	630	CPU Seconds
Turbulent Blunt	1542	CPU Seconds for 1000 sweeps

11. Results and Discussion

The results of the calculations are available on appendix c. The tabular results include the values of pressure, velocities, enthalpy, temperature mach number, density, turbulent kinetic energy and rate of turbulent dissipation. The values are given in 18 x 29 cells points.

Skin friction and heat transfer results are shown in Figures (5-11). Laminar and turbulent skin friction and Stanton number in wedge flow show improvement compared to the results reported by Yukselen (Ref. 10). The lines are smoother and the oscillations at the end were eliminated. Basically the magnitudes are similar to those in Ref. 10.

Laminar blunt values are similar except near the beginning. The beginning, as expected in blunt zone, creates higher rates of heat transfer. Even though the blunt geometry used is a multi-wedge shape it should predict the correct values except for the stagnation point itself.

Turbulent blunt skin friction has different behavior. It has a very large value at the first point and then undershoots to values that are smaller than for wedge. It should also be kept in mind that the convergence of this case wasn't very successful.

12. Conclusions and Recommendations

1. PHOENICS was found to be a friendly code for simulating complicated mixed heat transfer fluid dynamics problems.

2. Derivation of heat transfer properties to a vane solid wall in laminar and turbulent flow has been installed in the code. It can be used for predictions of heat transfer rate in both cold and hot gas flow.

3. Two features have been added to the code in NPS: The restart option and the use of initial field, make it possible to simulate time dependent processes and solve the temperature variation in the vane itself.

LIST OF REFERENCES

1. Baldwin and McCormak, "Numerical Solution of the Interaction of a Strong Shock Wave with a Turbulent Boundary Layer," AIAA Paper 74-558, AIAA 7th Fluid and Plasma Dynamics Conference Palo Alto, Calif., June 17-19, 1974.

2. Shang, J. S., Hankey, W. L., and Law, C. H., "Numerical Simulation of Shock Wave-Turbulent Boundary Layer Interaction," AIAA Paper, 76-95, AIAA 14th Aerospace Sciences Meeting, Washington, D.C., Janaury 1976.

3. Gunton, M. C., Rosten, H. L., and Spalding, D. B., Phoenics Instruction Manual, Spring 1983, CHAM Co, London.

4. White, Frank M., Viscous Fluid Flow, Mc Graw-Hill, Inc. 1974.

5. Launder, B., and Spalding, D. B., Lectures in Mathematical Models of Turbulence, Department of Mechanical Engineering Imperial College of Science and Technology, London, England, Academic Press, 1972.

6. Shapiro, Ascher H., The Dynamics and Thermodynamics of Compressible Fluid Flow, Volume, Department of Mechanical Engineering, Massachussetts Institute of Technology, Newyork, The Rolald Press Company.

7. Schlichting, Hermann, Engineering University of Braunschweig Newyork St. Louis, Boundary Layer Theory Mc Graw Hill Book Company, San Francisco Toronto, 1968.

8. Lin, C. C., Turbulent Flows and Heat Transfer, Princeton, New Jersey Princeton, New Jersey Princeton University Press, 1968.

9. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Mc Graw Hill, Inc., 1980.

10. Yukselen, A., Heat Transfer Modeling of Thrust Vector Control, Msc. Thesis, Naval Postgraduate School 1986.

11. Spalding D. B., "A General Computer Code for Two-Dimensional Elliptic Flows," Imperial College, London, 1977.

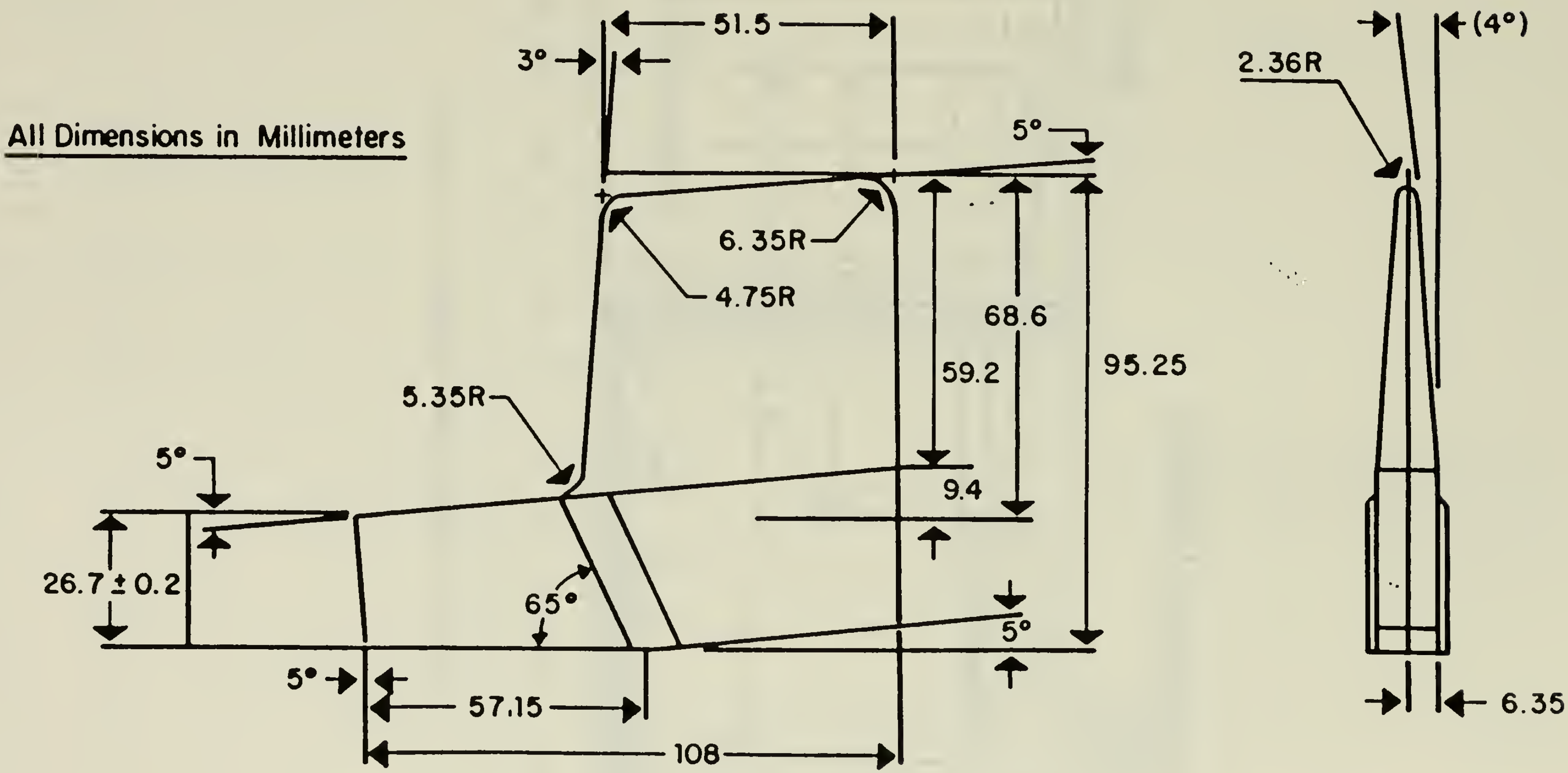

Figure 1: NWC Jet vane configuration.

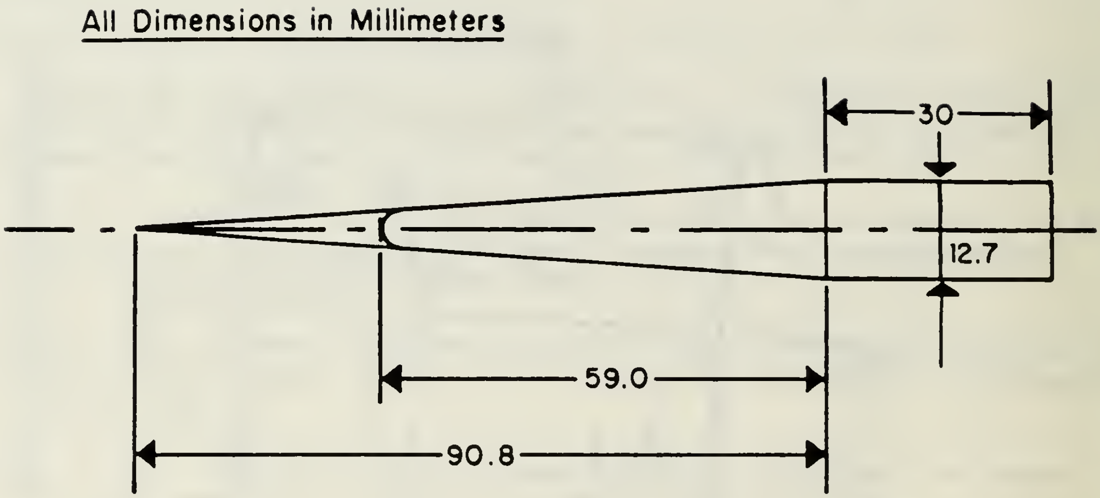

Figure 2: NWC Jet Vane Approximation

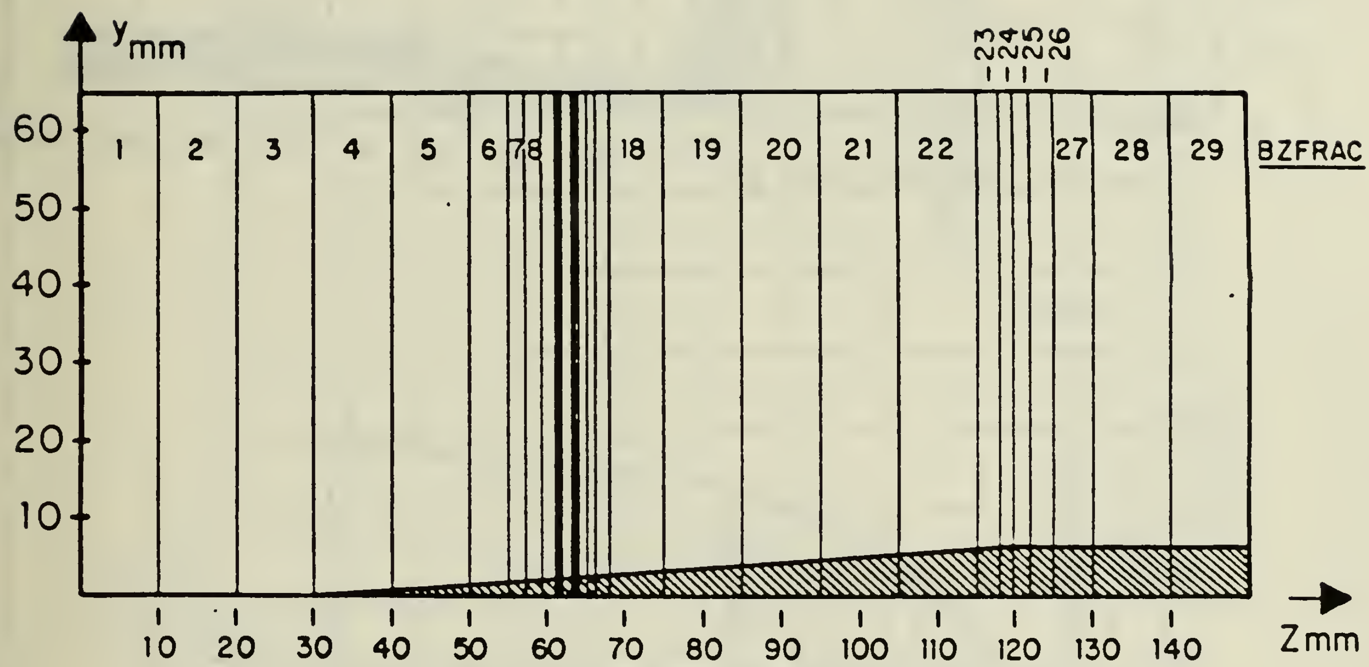

Figure 3: Wedge vane domain and grid

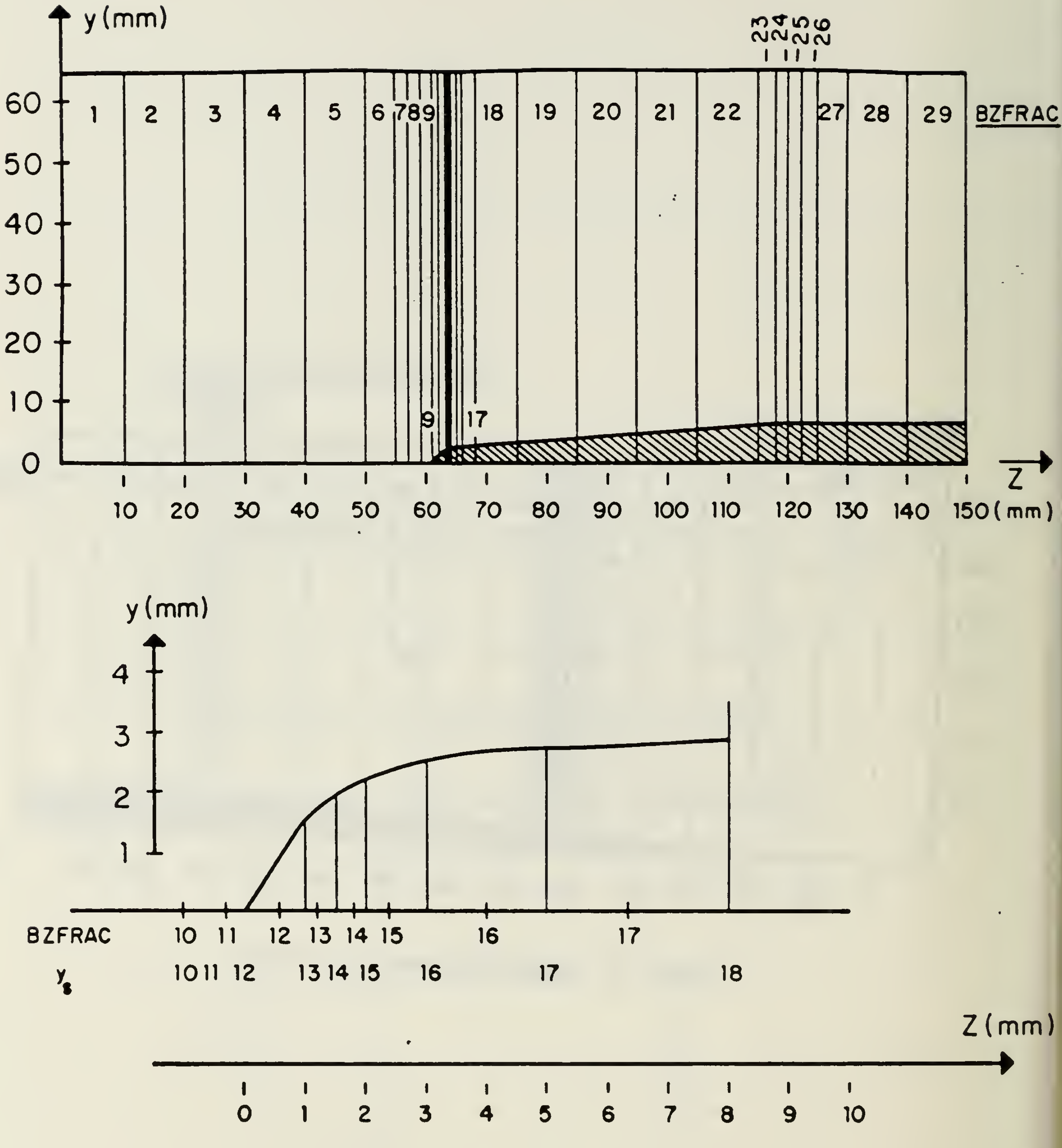

Figure 4: Blunt vane domain and grid.

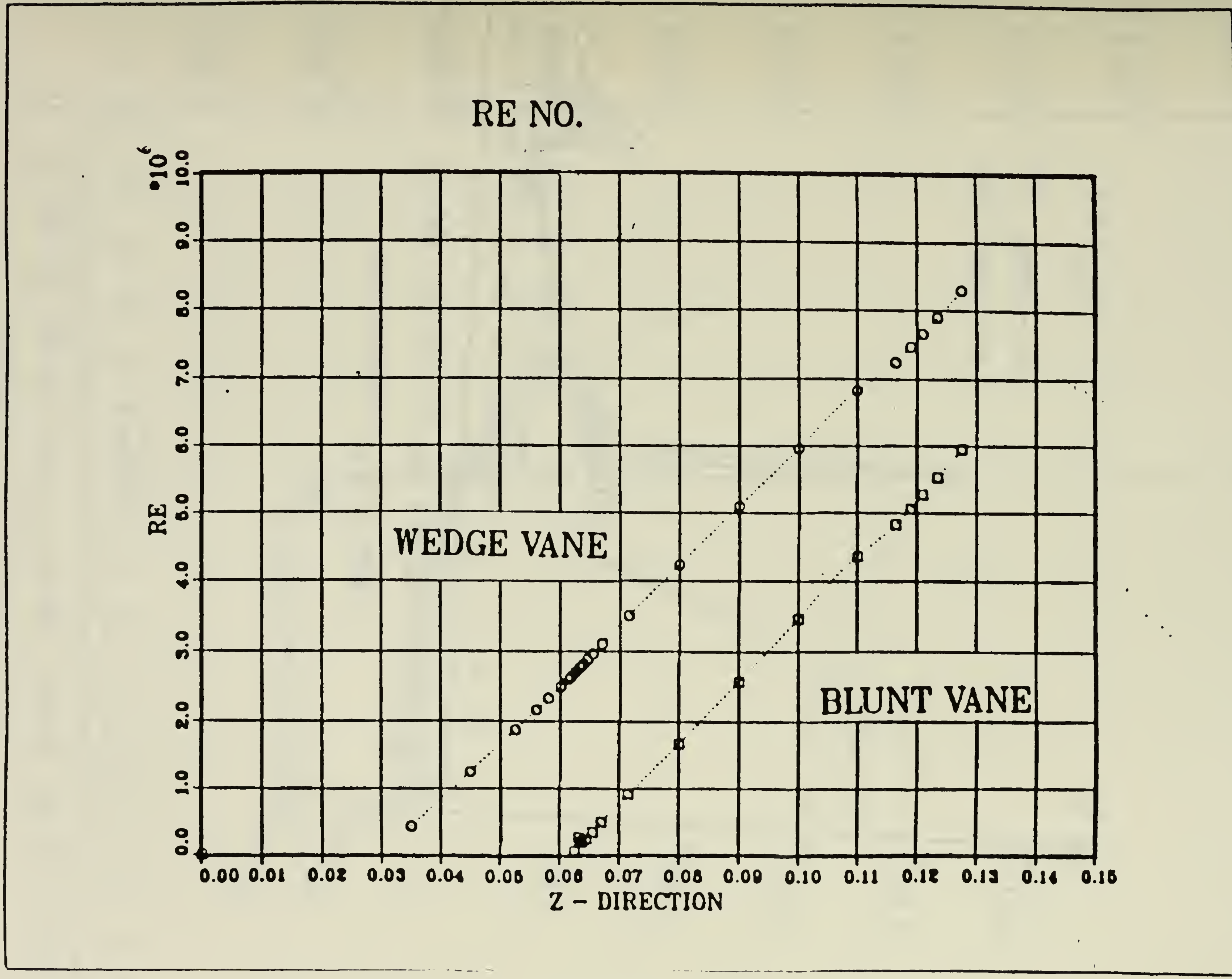

Figure 5: R'_{e_x} No. along the Vane

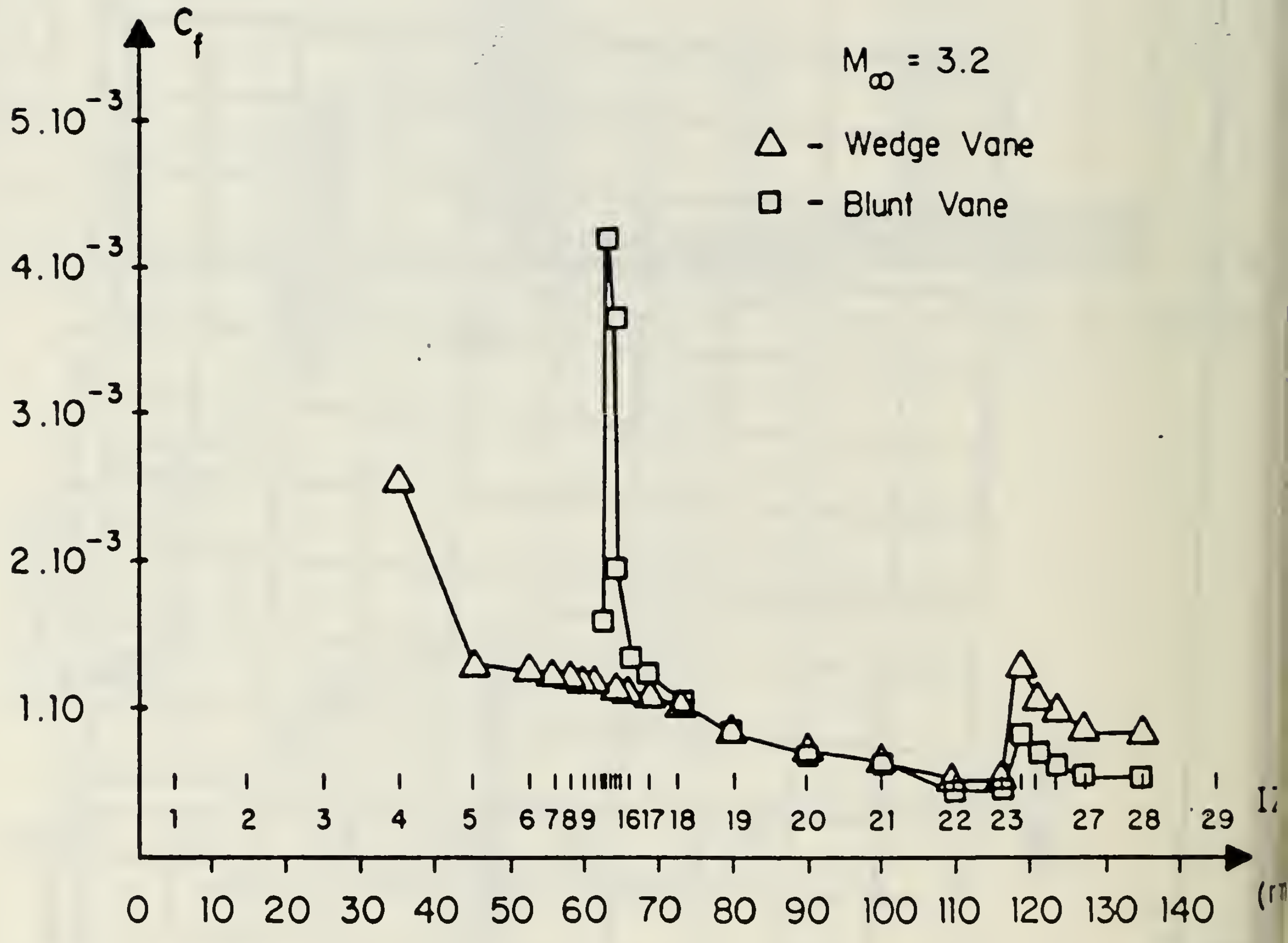

Figure 6: C_f in Laminar flow.

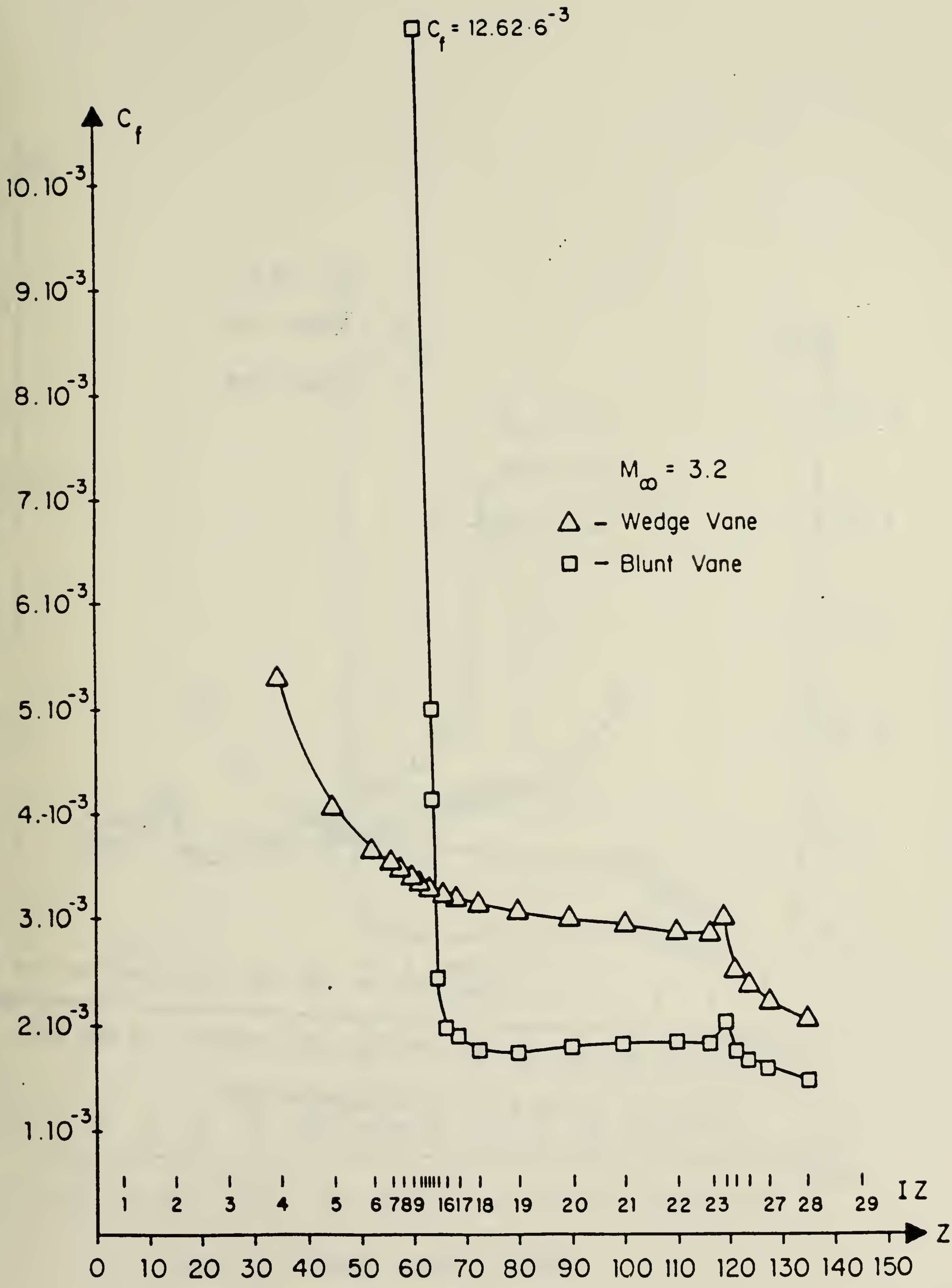

Figure 7: C_f in Turbulent flow

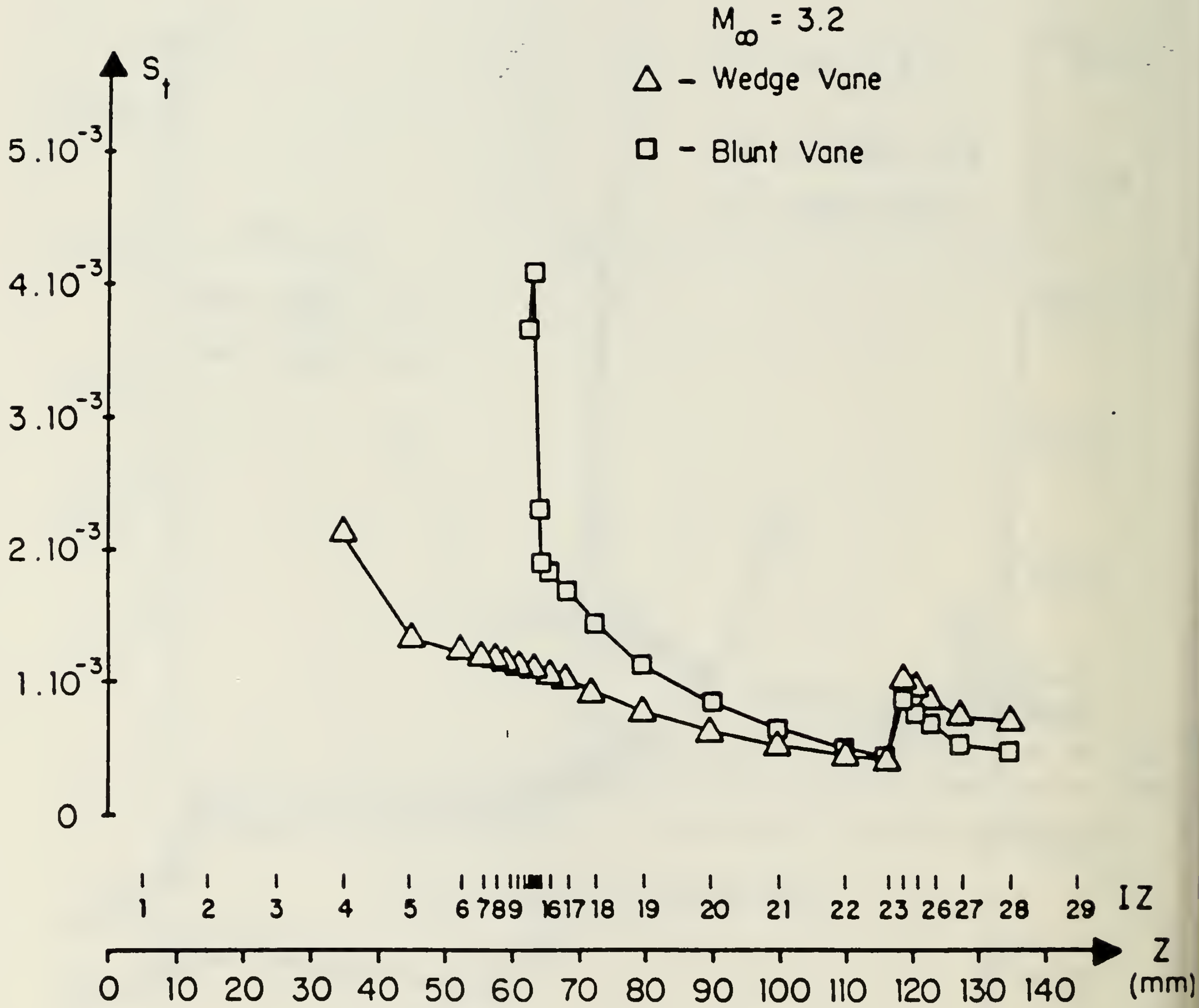

Figure 8: S_t in Laminar flow

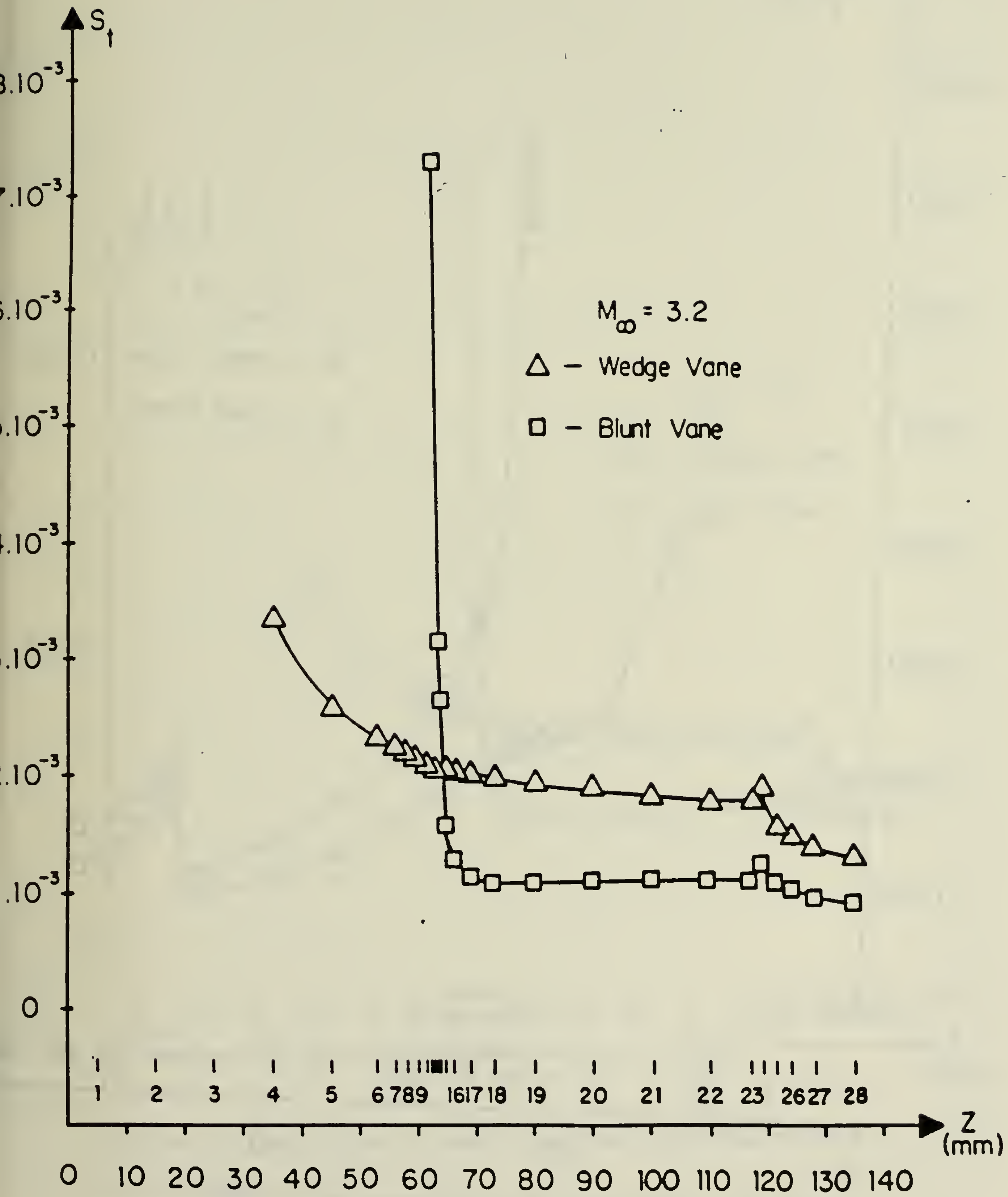

Figure 9: Turbulent stanton number.

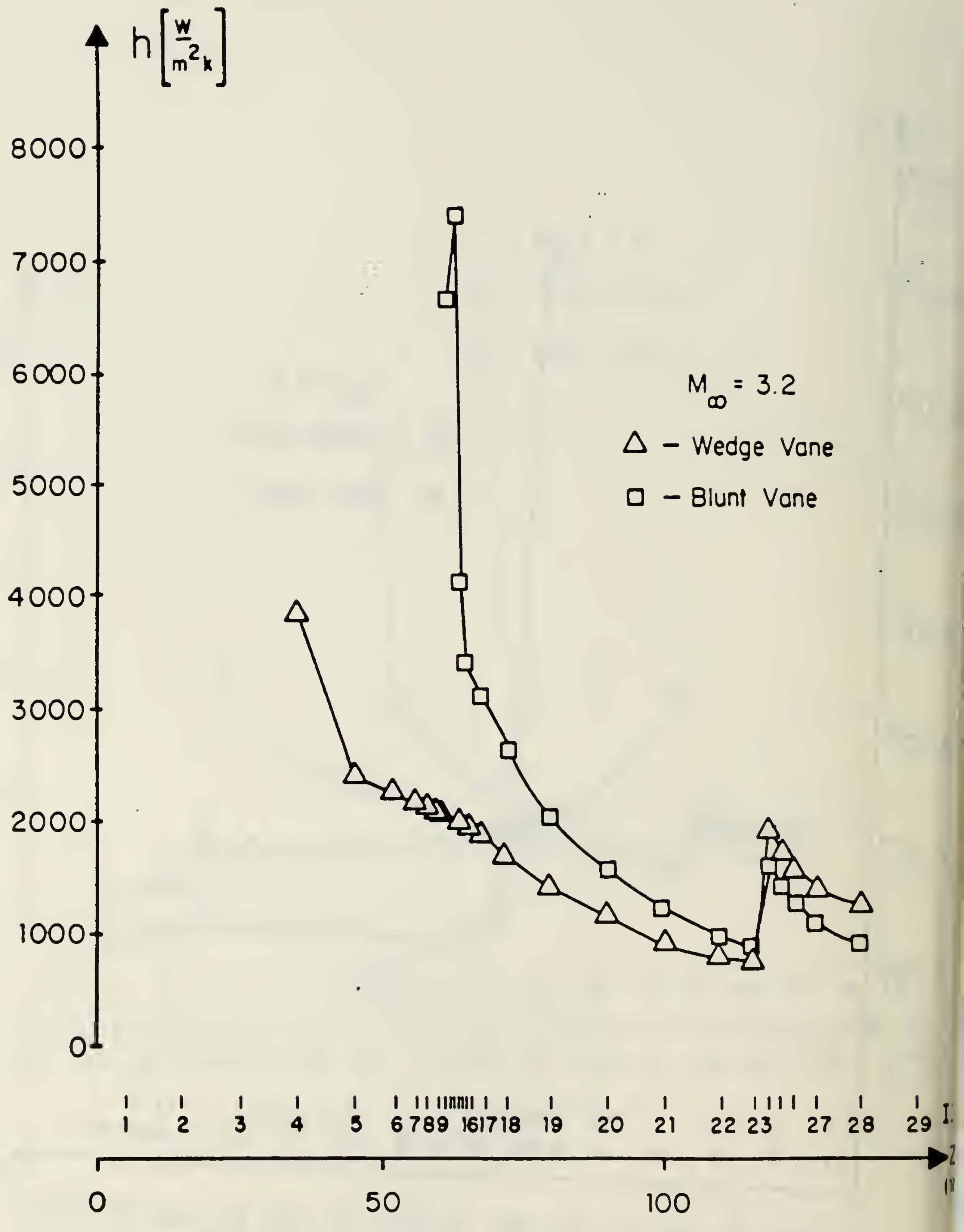

Figure 10: Coefficient of heat convection in laminar flow.

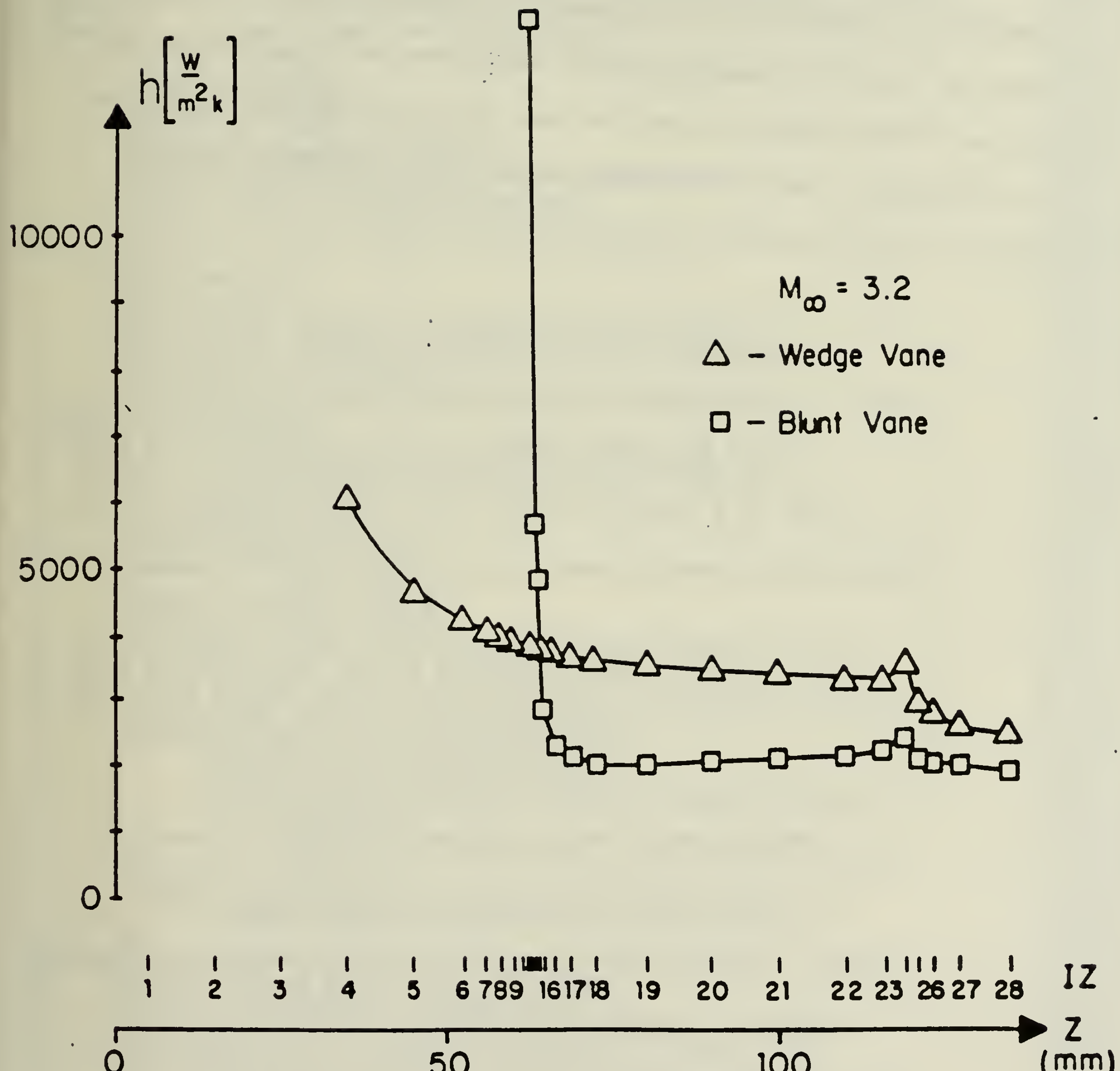

Figure 11: Coefficient of heat convection in turbulent flow.

Appendix A

Satellite Listing

Two subroutines SATELLITE and GROUND had to be changed and improved. The full list is enclosed in Appendix A and B.

VAN4SAT and VANTSAT are the laminar and turbulent SATELLITE subroutines, the first has the blunt geometry and the second has the wedge (it can be changed easily from wedge to blunt and vice versa) VAN4GRD and VANTGRD are exactly the same. They are the GROUND subroutines, VANTGRD is given in Appendix B.

```
C$DIRECTIVE**SATLIT   AMI LEITNER
C     LAMINAR SOLUTION FOR NWC5   NY=18 NZ=29 YN=GTH
C     LECSAT CONVERTED TO DIAMSAT
C    *FILE NAME: MODBFCST.FTN
C    *ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS
C     FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984
C     VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS.
C    *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C     WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C     (SUMMER 1984).
C    *AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
C     SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C-----------------------------------------------------------------
CHAPTER 1   COMMON BLOCKS AND USER'S DATA.
C-----------------------------------------------------------------
      INCLUDE (CMNGUS)
       INCLUDE (CMNGRF)
      INCLUDE (GUSSEQ)
      COMMON/CPI/IPWRIT,IDUM(243)
      DIMENSION GDTAPE(3),DFAULT(4)
      DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
      LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
      INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
      INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
     &C3,C4
      REAL NORTH,LOW
      LOGICAL BFC
      EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
      EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)
      EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))
      EQUIVALENCE (LOGIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
C$DIRECTIVE**CMNBF1$$
C  THIS FILE CONTAINS SATELLITE COMMON BLOCKS FOR BFC'S
C  F1 MUST BE DIMENSIONED TO GREATER THAN OR EQUAL TO
C (NX+NY+17*NZ+24*NX*NY+6*(NX+1)*(NY+1)+6*ND). THE VALUE
C  OF THE DIMENSION MUST BE SET AS NBFC IN GROUP 6 OF SATLIT.
      COMMON/F0B/F1(5000)
      COMMON/CIB/ND/CIC/KOORD
      COMMON/CID/KDBGG,KDBGMF,KDBGCD,KDBIND,KDBMFX,KDBCDT,KDBPCS,
     &            KDBGUV,KDBGPV
      COMMON/CIE/KDBGS,KDBINS
      COMMON/CIF/IGEN/CIG/NCART
C  THE FOLLOWING ARRAYS MUST BE EXACTLY DIMENSIONED FOR NXP1,
C  NYP1 AND NZP1, BUT MAY BE OVER DIMENSIONED FOR ND.
C  THE BFRAC ARRAYS MUST BE DIMENSIONED TO ALLOW FOR SETTINGS
C  IN SATLIT, THEY MAY BE OVER DIMENSIONED.
      COMMON/CRA/XW(19,30,1)/CRB/XE(19,30,1)
     &      /CRC/YS(2,30,1)/CRD/YN(2,30,1)
     &      /CRE/ZL(2,19,1)/CRF/ZH(2,19,1)
     &      /CRG/RCON/CRH/DARCY/CRI/BXFRAC(99)/CRJ/BYFRAC(99)
     &      /CRK/BZFRAC(99)
      COMMON/CLA/STORSA(6),STORWD(6),STORP,STORPE,STORPN,
     &            STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOCRN
      COMMON/CLC/BFPLOT
      LOGICAL STORP,STORPE,STORPN,STORPH,STOR1,STOR2,STOR3,
     &        STORSA,STORWD,STOUNV,PRTBFC,BFPLOT,STOCRN
C     END
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C     GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
      COMMON/GRAF1/PHI1(1) /GRAF2/PHI2(1)
C     POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...
      DIMENSION PE(1,1,1),PN(1,1,1),PH(1,1,1),PC(1,1,1)
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
C     USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      EQUIVALENCE(RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23))
     1,(GPR,RE(24)),(TW,RE(25)),(GEMU1,RE(26)),(JEMU1,INTGR(1))
C     USER PLACES HIS DATA STATEMENTS HERE.
      DATA NLSP,NISP,NRSP/1,1,1/
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
```

```
C--------------------------------------------------------------------
CHAPTER 2   SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C--------------------------------------------------------------------
C      PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C      STATEMENTS OF THIS CHAPTER.
       DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
      & 0.,1.,2.,3.,4.,5.,6.,7. /
       DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
      &C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
       DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
       DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
       DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
       DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
       DATA NLDATA,NIDATA,NRDATA/309,194,421/
       DATA NLCREG,NTCVRG/60,350/
       DATA TITPP,TITC1,TITC2,TITC3/3HRHO,4HMACH,4HTEMP,4HCFST/
       CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C------------------------------READ DEFAULT FILE IF BLOCKDATA ABSENT
          IF(INTGR1(29).NE.10) GO TO 2
       CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.           )
       GO TO 3
     2 CALL DEFLT
CD   2 CALL TAPES(1,DFAULT,4,2,4*NRDATA)
CD     CALL DATAIO(RD,1)
       CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
     3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.       )
       LOGIC(89)=.TRUE.
C--------------------------------------------------------------------
CHAPTER 3   DEFINE DATA FOR NRUN RUNS.
C--------------------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C--- GROUP  41MULTI-RUNS : RUN(1-30)<.T.,29*.F.>
C
       RUN(1)=.FALSE.
C  NOTE: ALL RUNS ARE DEACTIVATED AT THIS POINT - USER SHOULD
C  ====  SWITCH ON ONE ONLY OF RUNS 1-4 IN NEXT STATEMENT.
       RUN(1)=.TRUE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
         DO 10 IRUN=1,30
           IF(.NOT.RUN(IRUN)) GO TO 10
       NRUN=NRUN+1
       LSTRUN=IRUN
    10 CONTINUE
         DO 999 IRUN=1,LSTRUN
           IF(.NOT.RUN(IRUN)) GO TO 999
       INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C---   ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C      TO 0.0, UNLESS OTHERWISE INDICATED.
C      E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C      THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C      INDICATED, E.G. VARIABLE<.T.>, OR  VARIABLE<.F.>.
C
C---   RUN1
C------------------    ---------------    -------------------
C---   GROUP 1. FLOW TYPE :
C      PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
C------------------    ---------------    -------------------
C---   GROUP 2. TRANSIENCE :
C      STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C      TLAST<1.E10>,TFRAC(1-30)<30*1.>
C      SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
C      CALL GRDPWR(0,NT,TLAST,POWER)
C------------------    ---------------    -------------------
C---   GROUP 3. X-DIRECTION :
C      NX<1>,XULAST<1.0>,XFRAC(1-30)
C      SERVICE SUBROUTINE FOR POWER-LAW GRID:
C      CALL GRDPWR(1,NX,XULAST,POWER)
C------------------    ---------------    -------------------
```

```
C---    GROUP 4. Y-DIRECTION :
C       NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C       SERVICE SUBROUTINE FOR POWER-LAW GRID:
C       CALL GRDPWR(2,NY,YVLAST,POWER)
        NY=18
C------------------    ---------------    -------------------
C---    GROUP 5. Z-DIRECTION :
C       NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C       SERVICE SUBROUTINE FOR POWER-LAW GRID:
C       CALL GRDPWR(3,NZ,ZWLAST,POWER)
        NZ=29
C------------------    ---------------    -------------------
C---    GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID :
C                 --- MOVING GRID :
C       MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2M1T
C-------------------------------------------------------------
C                 --- BODY-FITTED GRID ---
C       BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
C       BXFRAC(1-NX)<1.0,NXM1*0.0>
C       BYFRAC(1-NY)<1.0,NYM1*0.0>
C       BZFRAC(1-NZ)<1.0,NZM1*0.0>
C       SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1
C       ONLY):
C          CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
C       XE(1-NYP1,1-NZP1,1-ND)<(NYP1*NZP1*ND)*1.0>,
C       XW(1-NYP1,1-NZP1,1-ND),
C       YN(1-NXP1,1-NZP1,1-ND)<(NXP1*NZP1*ND)*1.0>,
C       YS(1-NXP1,1-NZP1,1-ND),
C       ZH(1-NXP1,1-NYP1,1-ND)<(NXP1*NYP1*ND)*1.0>,
C       ZL(1-NXP1,1-NYP1,1-ND),STORSA(1-6)<6*.F.>,STORWD(1-6)<6*.F.>,
C       STORP<.F.>,STORPE<.F.>,STORPN<.F.>,STORPH<.F.>,STOUNV<.F.>,
C       PRTBFC<.F.>,DARCY,BFPLOT<.F.>
C       CYCLIC BOUNDARY CONDITIONS ARE DEFAULTED INACTIVE ;
C       TO ACTIVATE THEM AT SELECTED IZ SLABS USE SERVICE SUBROUTINE:
C          CALL XCYIZ(IZ,.TRUE.)
C       SERVICE SUBROUTINE TO DEACTIVATE CURVATURE TERMS IN U, V
C       AND W EQUATIONS ASSOCIATED WITH CURVATURE OF IX, IY, IZ
C       GRID LINES RESPECTIVELY:
C          CALL UCURVE(IZ,.FALSE.)
C          CALL VCURVE(IZ,.FALSE.)
C          CALL WCURVE(IZ,.FALSE.)
C       NCART<1>
C      *WARNINGS!!!!!!
C       -------------
C         A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
C            AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
C            COMPONENTS.
C         B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C         C) MOVING GRID,TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
C            AVAILABLE WITH BFC OPTION.
C         D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
C            WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
C         E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.
C      *NOTES
C       -----
C         A) THE STANDARD VELOCITY-FIELD PRINTOUT FOR THE
C            VELOCITY RESOLUTES IS ACTIVATED IN THE USUAL
C            WAY. AN ADDITIONAL OPTION EXISTS FOR PRINTING THE
C            CARTESIAN VELOCITY-COMPONENTS WHICH MAY BE
C            ACTIVATED BY SETTING THE FOLLOWING LOGICALS:
C                 STOVAR(U2)=.T. FOR U-COMPONENT (CARTESIAN)
C                 STOVAR(V2)=.T. FOR V-COMPONENT (CARTESIAN)
C                 STOVAR(W2)=.T. FOR W-COMPONENT (CARTESIAN)
C            SIMILARLY PRINTOUT OF NON-ORTHOGONAL VELOCITY
C            COMPONENTS MAY BE ACTIVATED AS FOLLOWS:
C                 STOVAR(C4)=.T. FOR U-COMPONENT (NON-ORTHOG)
C                 STOVAR(H3)=.T. FOR V-COMPONENT (NON-ORTHOG)
C                 STOVAR(21)=.T. FOR W-COMPONENT (NON-ORTHOG)
C         B) BFC (TO ACTIVATE THE BFC OPTION), IGEN (THE CODE FOR METHOD
C            OF GRID SPECIFICATION), ND (NUMBER OF SUB-DOMAINS) AND
C            NBFC (THE F1 ARRAY DIMENSION), MUST BE SET BEFORE
C            "STANDARD BFC SECTION 2".                      ======
```

```
C           ALL OTHER BFC DATA MUST BE SET AFTER "STANDARD BFC
C           SECTION 2.                           =====
C        C) NXP1, NYP1, NZP1 STORE NX+1, NY+1, NZ+1; THESE ARE
C           AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
C        D) FOR IGEN=1 USE BXFRAC,BYFRAC & BZFRAC IN PLACE OF
C           XFRAC,YFRAC & ZFRAC.
C------------------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 STARTS:
C     DEFAULT SETTINGS:
      NCART=10
      BFC=.TRUE.
      IGEN=1
      ND=1
      NBFC=5000
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 ENDS.:
C    *USER SETS BFC, IGEN, ND AND NBFC HERE:
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 STARTS:
      CALL SB4I(NXP1,NX+1,NYP1,NY+1,NZP1,NZ+1,I,0)
      IF(BFC) CALL BFCDFT(NBFC,XE,XW,YN,YS,ZH,ZL,ND,NXP1,NYP1,
     &                    NZP1,NZ)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 ENDS.
C    *USER SETS ALL OTHER BFC VARIABLES HERE:
C    *USING NONIFORM GRID 1-8
      GTH=65.E-3
      GTL=150.E-3
      GBETA=4.
      GBETA=GBETA*3.1415927/180
      GTAB=TAN(GBETA)
      DELMAX=2.E-3
      GNBL=5.
      GPWR=2.
      DO 64 IY=1,5
   64 BYFRAC(IY)=(FLOAT(IY)/GNBL)**GPWR*DELMAX/GTH
      BYFRAC(6)=BYFRAC(5)+3.E-3/GTH
      DEL=(1.-BYFRAC(6))/(FLOAT(NY)-GNBL-1)
      DO 65 IY=7,NY
   65 BYFRAC(IY)=BYFRAC(IY-1)+DEL
C----------ZZ-----------
      BZFRAC(1)=10.E-3
      DO 66 IZ=2,5
   66 BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
      BZFRAC(6)=BZFRAC(5)+5.E-3
      DO 67 IZ=7,9
   67 BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
      DO 68 IZ=10,10
   68 BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
      DO 77 IZ=11,14
   77 BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
      DO 78 IZ=15,15
   78 BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
      BZFRAC(16)=BZFRAC(15)+2.E-3
      BZFRAC(17)=BZFRAC(16)+3.E-3
      BZFRAC(18)=BZFRAC(17)+5.E-3
      DO 69 IZ=19,22
   69 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
      BZFRAC(23)=BZFRAC(22)+3.E-3
      BZFRAC(24)=BZFRAC(23)+2.E-3
      BZFRAC(25)=BZFRAC(24)+2.E-3
      BZFRAC(26)=BZFRAC(25)+3.E-3
      BZFRAC(27)=BZFRAC(26)+5.E-3
      DO 71 IZ=28,NZ
   71 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
      DO 72 IZ=1,NZ
   72 BZFRAC(IZ)=BZFRAC(IZ)/GTL
      CALL DOMAIN(1,1,NX,1,NY,1,NZ)
      DO 61 IX=1,NXP1
      DO 62 IY=1,NYP1
      ZL(IX,IY,1)=0.0
   62 ZH(IX,IY,1)=GTL
      DO 63 IZ=1,NZP1
      YN(IX,IZ,1)=GTH
```

```
  63  YS(IX,IZ,1)=0.0
C  YS(IX,13,1) SHOULD COME AFTER
      DO 662 IZ=16,25
CCC   DO 662 IZ=5,25
 662  YS(IX,IZ,1)=(BZFRAC(IZ-1)-BZFRAC(3))*GTAB*GTL
      DO 663 IZ=13,15
      GZ12=(BZFRAC(IZ-1)-BZFRAC(11))*GTL
  663 YS(IX,IZ,1)=SQRT(YS(IX,16,1)*GZ12*2.-GZ12**2)
      DO 664 IZ=26,NZ
 664  YS(IX,IZ,1)=YS(IX,25,1)
  61  CONTINUE
      STORSA(IFIX(LOW))=.TRUE.
      STORSA(IFIX(HIGH))=.TRUE.
      STORSA(IFIX(SOUTH))=.TRUE.
      STORWD(IFIX(SOUTH))=.TRUE.
      STORP=.TRUE.
      PRTBFC=.TRUE.
CDAR  DARCY=1.E10
C------------------    ---------------    -------------------
C---  GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
C    *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
C     CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
C     IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE'= EAST,
C     WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
C     OVER REGION IXF,...IZL.
C    *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
C     PC(NX,NY,NZ) ABOVE.
C    *FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
C     THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
C     AUTOMATICALLY ZEROED.
C    *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
C     XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
C    *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
C     BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
C     IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
C     IN THIS CASE, THE USER  M U S T   SET  A L L  ELEMENTS OF
C     ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
C     CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
C     ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
C     'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
C    *CONPOR  M U S T  N O T  BE USED IN CONJUNCTION WITH EXPLICIT
C     SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
C------------------    ---------------    -------------------
C---  GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
C     SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
C     USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
C     P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
      SOLVAR(P1)=.TRUE.
      SOLVAR(PP)=.TRUE.
      SOLVAR(V1)=.TRUE.
      SOLVAR(W1)=.TRUE.
      SOLVAR(H1)=.TRUE.
      SOLVAR(KE)=.TRUE.
      SOLVAR(EP)=.TRUE.
      STOVAR(V2)=.TRUE.
      STOVAR(W2)=.TRUE.
      STOVAR(C1)=.TRUE.
      STOVAR(C2)=.TRUE.
      STOVAR(C3)=.TRUE.
C------------------    ---------------    -------------------
C---  GROUP 9. VARIABLE LABELS :
C     TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
C   . 2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
C   . 2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H****>
      TITLE(C1)=TITC1
      TITLE(C2)=TITC2
      TITLE(C3)=TITC3
      TITLE(PP)=TITPP
C------------------    ---------------    -------------------
C---  GROUP 10 PROPERTIES:
C     IRHO1<1>,IRHO2<1>,RHO1<1.0>,RHO2<1.0>,
C     ARHO1<1.0>,BRHO1<1.0>,CRHO1<1.0>
```

```
C       IEMU1<1>,EMU1<1.0>,EMULAM<1.E-10>
C       IHSAT,H1SAT,H2SAT,PSATEX<1.0>
C       SIGMA(1-25)<1.0,2.0,1.,1.E10,1.,1.E10,1.,1.E10,
C       4*1.0,1.314,1.0,1.E10,10*1.0>
        IRHO1=-1
        PTOT=55.E5
        TOT=555.55
        RAIR=287.
        GAMA=1.35
        CP=RAIR/(1-1/GAMA)
        TW=323.
        HWALL=TW*CP
        HTOT=CP*TOT
        RHTOT=PTOT/TOT/RAIR
        LOGIC(87)=.TRUE.
        ARHO1=RHTOT/PTOT**(1/GAMA)
        BRHO1=1./GAMA
C   TURBULENT OR LAMINAR
        IEMU1=2
C       IEMU1=-1
        JEMU1=IEMU1
        EMU1=1.E-5
        EMULAM=EMU1
        GEMU1=EMU1
        GPR=.7
        SIGMA(24)=GPR
        SIGMA(14)=.9
C------------------    ---------------    -------------------
C---    GROUP 11 INTER-PHASE TRANSFER PROCESSES :
C       ICFIP,CFIPS,IMDOT,CMDOT,CA1I<1.E6>,CA2I<1.E6>
C------------------    ---------------    -------------------
C---    GROUP 12 SPECIAL SOURCES :
C       ISPCSO(1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF
C------------------    ---------------    -------------------
C---    GROUP 13 INITIAL FIELDS :
C       FIINIT(1-25)<25*1.E-10>
C     MACH NO. OF FREE STREAM
        GMACH=3.2
        A=1+(GAMA-1)/2*GMACH**2
        TE=TOT/A
        RHE=RHTOT/A**(1/(GAMA-1))
        PSTAT=PTOT/A**(GAMA/(GAMA-1))
        RHO1=ARHO1*PSTAT**BRHO1
        SONIC=SQRT(GAMA*RAIR*TE)
        WIN=SONIC*GMACH
        RKEIN=0.01*WIN**2
        EPIN=0.16*RKEIN**1.5/GTH/2.
        FIINIT(W1)=WIN
        FIINIT(P1)=PSTAT
        FIINIT(H1)=HTOT
        FIINIT(KE)=RKEIN
        FIINIT(EP)=EPIN
C------------------    ---------------    -------------------
C---    GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C       ILOOP1,ILOOPN,XCYCLE<.F.>,PBAR,REGION(1-10)<10*.T.>
C      *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE..  THE USER SHOULD
C       SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
        DO 14 I=1,10
   14   REGION(I)=.FALSE.
C------------------    ---------------    -------------------
C---    GROUP 15 TO 24; REGIONS 1 TO 10
C---    ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE
C       USER, PREFERABLY BY WAY OF:-
C       CALL PLACE(IREGN,TYPE,IXF,IXL,IYF,IYL,IZF,IZL) &
C       CALL COVAL(IREGN,VARBLE,COEFF,VALUE)
        CALL PLACE(1,LOW,1,NX,1,NY,1,1)
C       CALL COVAL(1,M1,FIXFLU,WIN*RHE)
CDAR    CALL COVAL(1,M1,1.E-20,1.E+20*WIN*RHE)
        GCM=2*GAMA/WIN/(GAMA-1)
        GVM=PTOT*RHE/RHTOT
        CALL COVAL(1,M1,GCM,GVM)
        CALL COVAL(1,W1,ONLYMS,WIN)
```

```
      CALL COVAL(1,H1,ONLYMS,HTOT)
      CALL COVAL(1,KE,ONLYMS,RKEIN)
      CALL COVAL(1,EP,ONLYMS,EPIN)
      CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
C     CALL COVAL(2,M1,FIXVAL,PSTAT*0.)
      CALL COVAL(2,M1,1000*WIN*RHE/PSTAT,PSTAT)
      CALL COVAL(2,H1,ONLYMS,HTOT)
C   WALL ALONG THE VANE  IZ(11,NZ)
      GCM=EMU1/(.5*BYFRAC(1)*GTH)
      DY1=BYFRAC(1)*GTH
      GOEFF=EMU1/(0.5*DY1)
      GOEFH=EMU1/(0.5*DY1*SIGMA(24))
      CALL PLACE(3,SOUTH,1,NX,1,1,12,NZ)
C     CALL COVAL(3,W1,GOEFF,0.)
C     CALL COVAL(3,H1,GOEFH,HWALL)
      CALL COVAL(3,W1,WALL,0.)
      CALL COVAL(3,H1,WALL,HWALL)
      CALL COVAL(3,KE,WALL,0.)
      CALL COVAL(3,EP,WALL,0.)
C-----------------   ---------------    --------------------
C---  GROUP 25 GROUND STATION :
C     GROSTA<.F.>,NAMLST<.F.>
C    *NAMLST ACTIVATES NAMELIST IN GROUND.
      GROSTA=.TRUE.
C-----------------   ---------------    --------------------
C---  GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C     WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
      WHOLEP=.TRUE.
C-----------------   ---------------    --------------------
C---  GROUP 27 SWEEP AND ITERATION NUMBERS :
C     FSWEEP<1>,LSWEEP<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
C     LITER(1-25)<9*1,-1,15*1>
C     IVELF<1>,NVEL<1>,IVELL<10000>,
C     IKEF<1>,NKE<1>,IKEL<10000>,
C     IENTF<1>,NENT<1>,IENTL<10000>,
C     ICNCF<1>,NCNC<1>,ICNCL<10000>,
C     IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
C     IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
      LSWEEP=1201
      GSWP=LSWEEP
      FSWEEP=801
      LITER(PP)=20
      LITER(V1)=5
      LITER(W1)=5
C     LITHYD=2
C-----------------   ---------------    --------------------
C---  GROUP 28 TERMINATION CRITERIA :
C     ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
      ENDIT(1)=1.E-5
C-----------------   ---------------    --------------------
C---  GROUP 29 RELAXATION  :
C     RLXP<1.>,RLXPXY<1.>,RLXPZ<1.>,RLXRHO<1.>,RLXMDT<1.>,
C     DTFALS(3-25)<23*1.E10>
      DTFALS(W1)=1.E-5
      DTFALS(V1)=1.E-5
      DTFALS(KE)=1.E-5
      DTFALS(EP)=1.E-6
      RLXP=.3
C-----------------   ---------------    --------------------
C---  GROUP 30 LIMITS :
C     VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
C     TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
C     EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
      EPSMAX=1.E13
C-----------------   ---------------    --------------------
C---  GROUP 31 SLOWING DEVICES : SLORHO<1.>,SLOEMU<1.>
      SLORHO=.2
C-----------------   ---------------    --------------------
C---  GROUP 32 PRINT-OUT OF VARIABLES :
C     PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
      PRINT(C1)=.TRUE.
      PRINT(C2)=.TRUE.
```

```
      PRINT(C3)=.TRUE.
      PRINT(PP)=.TRUE.
C------------------    --------------    ------------------
C---  GROUP 33 MONITOR PRINT-OUT :
C     IXMON<1>,IYMON<1>,IZMON<1>,NPRMON<1>,NPRMNT<1>
      NPRMON=10
      IYMON=2
      IZMON=12
C------------------    --------------    ------------------
C---  GROUP 34 FIELD PRINT-OUT CONTROL :
C     NPRINT<100>,NTPRIN<100>,NXPRIN<1>,NYPRIN<1>,NZPRIN<1>,
C     IZPRF<1>,ISTPRF<1>,IZPRL<10000>,ISTPRL<10000>
C     NUMCLS<10>,KOUTPT
      NPRINT=LSWEEP
C------------------    --------------    ------------------
C---  GROUP 35 TABLE CONTROL :
C     TABLES<.F.>,NTABLE,NTABVR,LINTAB,NPRTAB,NMON,
C     ITAB(1-8),MTABVR(1-8)
C------------------    --------------    ------------------
C     GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C     MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C---  GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C     IZPR1<1>,IZPR2<1>,ISTPR1<1>,ISTPR2<1>
C------------------    --------------    ------------------
C---  GROUP 37 DEBUG SWEEP AND SUBROUTINES :
C     KEMU,KMAIN,KINDEX,KGEOM,KINPUT,KSODAT,KCOMPF,KSORCE,
C     KSOLV1,KSOLV2,KSOLV3,KCOMPP,KADJST,KFLUX,KSHIFT,KDIF,
C     KCOMPU,KCOMPV,KCOMPW,KCOMPR,KWALL,KDBRHO<-1>,KDBEXP,KDBMDT
C     KDBGEN
C------------------    --------------    ------------------
C---  GROUP 38 MONITOR,TEST,AND FLAG :
C     MONITR<.F.>,FLAG<.F.>,TEST<.T.>,KFLAG<1>
C     END OF MAINTENANCE-ONLY SECTION
C------------------    --------------    ------------------
C---  GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C     IERRP<1000>,RESREF(1,3-24)<25*1.>,RESMAP<.F.>,
C     RESID(1-25)<2*.F.,23*.T.>,KOUTPT
      RESREF(1)=WIN*RHE
      RESREF(7)=WIN*RESREF(1)
      RESREF(5)=WIN*RESREF(1)*0.1
      RESREF(H1)=HTOT*RESREF(1)
      RESREF(KE)=RKEIN*RESREF(1)
      RESREF(EP)=EPIN*RESREF(1)
      IERRP=LSWEEP/20
      KOUTPT=LSWEEP/20
C------------------    --------------    ------------------
C---  GROUP 40 SPECIAL DATA : LOGIC(1..10),INTGR(1..10),RE(21..30),
C     NLSP<1>,NISP<1>,NRSP<1>,SPDATA<.F.>,LSPDA(1),ISPDA(1),RSPDA(1)
C     USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C     TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C     SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C     PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C     ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C--------------------------------------------------------------
C---  GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>,RESTRT<.F.>,KINPUT
      SAVEM=.TRUE.
      BFPLOT=.TRUE.
      RESTRT=.TRUE.
C------------------    --------------    ------------------
C------------------    --------------    ------------------
C---  GROUP 43 GRAFFIC :
C     GRAPHS<.F.>,ORTHOG<.T.>,ANTSYM,NPRT<1>,ITITL<5*4H****>
C---  FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C     PHI1(NX*NY*NZ*NM)
C     PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)) , WHERE
C     NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C     IBLK=0 IF BLOCK=.FALSE.,=4 IF A 3D RUN,
C     =3 IF A 2D.YZ RUN.
C------------------    --------------    ------------------
          IF(IRUN.EQ.1) GO TO 900
  900 CONTINUE
C---  ALL RUNS
```

```
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 4 STARTS:
C-------------------------------------------------------------
C  WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...
          IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
          IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)
      IF(BFC) CALL WRTBFC(14,NBFC,XE,XW,YN,YS,ZH,ZL,
     &ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)
C     OLD PRACTICES RETAINED FOR REFERENCE:
C         IF(SPDATA) CALL SPCDAT(IRUN)
C         IF(BLOCK) CALL PORDAT(IRUN)
          IF(GRAPHS) CALL SORT(IRUN)
          IF(RESTRT) GO TO 902
        DO 901 INDVAR=1,25
          IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901
      CALL FLDDAT(IRUN)
      GO TO 902
  901 CONTINUE
  902 CALL DATAIO(WRT,10)
          IF(MONITR) CALL DATAIO(WRT,-6)
 999  CONTINUE
      STOP
      END
C***   IGEN=1 SO BFCXYZ NOT REQUIRED.
C***   COMMENT OUT BOTH VERSIONS.
C--------------------------------------------------------------
      SUBROUTINE BFCXYZ (NXP1,NYP1,NZP1)
      RETURN
      END
```

FILE: VAN4SAT FORTRAN A1

```
C$DIRECTIVE**SATLIT   AMI LEITNER
C      LAMINAR SOLUTION FOR NWC5   NY=18 NZ=29 YN=GTH
C      LECSAT CONVERTED TO DIAMSAT
C     *FILE NAME: MODBFCST.FTN
C     *ABSTRACT: SATELLITE MODEL MAIN PROGRAM. THIS VERSION IS
C      FOR USE WITH THE BODY-FITTED COORDINATE SCHEME (SUMMER 1984
C      VERSION) PROVIDED AS AN ATTACHMENT TO SPRING 1983 PHOENICS.
C     *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C      WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C      (SUMMER 1984).
C     *AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
C      SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C-----------------------------------------------------------------
CHAPTER 1   COMMON BLOCKS AND USER'S DATA.
C-----------------------------------------------------------------
      INCLUDE (CMNGUS)
       INCLUDE (CMNGRF)
      INCLUDE (GUSSEQ)
      COMMON/CPI/IPWRIT,IDUM(243)
      DIMENSION GDTAPE(3),DFAULT(4)
      DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
      LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
      INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
      INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
     &C3,C4
      REAL NORTH,LOW
      LOGICAL BFC
      EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
      EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)
      EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))
      EQUIVALENCE (LOGIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
C$DIRECTIVE**CMNBF1$$
C  THIS FILE CONTAINS SATELLITE COMMON BLOCKS FOR BFC'S
C  F1 MUST BE DIMENSIONED TO GREATER THAN OR EQUAL TO
C (NX+NY+17*NZ+24*NX*NY+6*(NX+1)*(NY+1)+6*ND). THE VALUE
C  OF THE DIMENSION MUST BE SET AS NBFC IN GROUP 6 OF SATLIT.
      COMMON/F0B/F1(5000)
      COMMON/CIB/ND/CIC/KOORD
      COMMON/CID/KDBGG,KDBGMF,KDBGCD,KDBIND,KDBMFX,KDBCDT,KDBPCS,
     &           KDBGUV,KDBGPV
      COMMON/CIE/KDBGS,KDBINS
      COMMON/CIF/IGEN/CIG/NCART
C  THE FOLLOWING ARRAYS MUST BE EXACTLY DIMENSIONED FOR NXP1,
C  NYP1 AND NZP1, BUT MAY BE OVER DIMENSIONED FOR ND.
C  THE BFRAC ARRAYS MUST BE DIMENSIONED TO ALLOW FOR SETTINGS
C  IN SATLIT, THEY MAY BE OVER DIMENSIONED.
      COMMON/CRA/XW(19,30,1)/CRB/XE(19,30,1)
     &      /CRC/YS(2,30,1)/CRD/YN(2,30,1)
     &      /CRE/ZL(2,19,1)/CRF/ZH(2,19,1)
     &      /CRG/RCON/CRH/DARCY/CRI/BXFRAC(99)/CRJ/BYFRAC(99)
     &      /CRK/BZFRAC(99)
      COMMON/CLA/STORSA(6),STORWD(6),STORP,STORPE,STORPN,
     &           STORPH,STOR1,STOR2,STOR3,STOUNV,PRTBFC,STOCRN
      COMMON/CLC/BFPLOT
      LOGICAL STORP,STORPE,STORPN,STORPH,STOR1,STOR2,STOR3,
     &        STORSA,STORWD,STOUNV,PRTBFC,BFPLOT,STOCRN
C     END
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C     GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
      COMMON/GRAF1/PHI1(1) /GRAF2/PHI2(1)
C     POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...
      DIMENSION PE(1,1,1),PN(1,1,1),PH(1,1,1),PC(1,1,1)
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
C     USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      EQUIVALENCE(RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23))
     1,(GPR,RE(24)),(TW,RE(25)),(GEMU1,RE(26)),(JEMU1,INTGR(1))
C     USER PLACES HIS DATA STATEMENTS HERE.
      DATA NLSP,NISP,NRSP/1,1,1/
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
```

```
C-----------------------------------------------------------------
CHAPTER 2   SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
C-----------------------------------------------------------------
C      PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C      STATEMENTS OF THIS CHAPTER.
       DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
     & 0.,1.,2.,3.,4.,5.,6.,7. /
       DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
     &C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
       DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
       DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
       DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
       DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
       DATA NLDATA,NIDATA,NRDATA/309,194,421/
       DATA NLCREG,NTCVRG/60,350/
       DATA TITPP,TITC1,TITC2,TITC3/3HRHO,4HMACH,4HTEMP,4HCFST/
       CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C-----------------------------READ DEFAULT FILE IF BLOCKDATA ABSENT
          IF(INTGR1(29).NE.10) GO TO 2
       CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.         )
       GO TO 3
    2 CALL DEFLT
CD  2 CALL TAPES(1,DFAULT,4,2,4*NRDATA)
CD     CALL DATAIO(RD,1)
       CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
    3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.     )
       LOGIC(89)=.TRUE.
C-----------------------------------------------------------------
CHAPTER 3   DEFINE DATA FOR NRUN RUNS.
C-----------------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C--- GROUP  41MULTI-RUNS : RUN(1-30)<.T.,29*.F.>
C
       RUN(1)=.FALSE.
C  NOTE: ALL RUNS ARE DEACTIVATED AT THIS POINT - USER SHOULD
C  ====  SWITCH ON ONE ONLY OF RUNS 1-4 IN NEXT STATEMENT.
       RUN(1)=.TRUE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
         DO 10 IRUN=1,30
           IF(.NOT.RUN(IRUN)) GO TO 10
       NRUN=NRUN+1
       LSTRUN=IRUN
   10 CONTINUE
         DO 999 IRUN=1,LSTRUN
           IF(.NOT.RUN(IRUN)) GO TO 999
       INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C---   ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C      TO 0.0, UNLESS OTHERWISE INDICATED.
C      E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C      THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C      INDICATED, E.G. VARIABLE<.T.>, OR  VARIABLE<.F.>.
C
C---   RUN1
C------------------    --------------    ------------------
C---   GROUP 1. FLOW TYPE :
C      PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
C------------------    --------------    ------------------
C---   GROUP 2. TRANSIENCE :
C      STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C      TLAST<1.E10>,TFRAC(1-30)<30*1.>
C      SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
C      CALL GRDPWR(0,NT,TLAST,POWER)
C------------------    --------------    ------------------
C---   GROUP 3. X-DIRECTION :
C      NX<1>,XULAST<1.0>,XFRAC(1-30)
C      SERVICE SUBROUTINE FOR POWER-LAW GRID:
C      CALL GRDPWR(1,NX,XULAST,POWER)
C------------------    --------------    ------------------
```

```
C---  GROUP 4. Y-DIRECTION :
C     NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C     SERVICE SUBROUTINE FOR POWER-LAW GRID:
C     CALL GRDPWR(2,NY,YVLAST,POWER)
      NY=18
C------------------      ---------------     -------------------
C---  GROUP 5. Z-DIRECTION :
C     NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C     SERVICE SUBROUTINE FOR POWER-LAW GRID:
C     CALL GRDPWR(3,NZ,ZWLAST,POWER)
      NZ=29
C------------------      ---------------     -------------------
C---  GROUP 6. MOVING GRID OR DISTORTED (BODY-FITTED) GRID :
C             --- MOVING GRID :
C     MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2M1T
C-----------------------------------------------------------------
C             --- BODY-FITTED GRID ---
C     BFC<.T.>,IGEN<1>,ND<1>,NBFC<5000>,KOORD,RCON
C     BXFRAC(1-NX)<1.0,NXM1*0.0>
C     BYFRAC(1-NY)<1.0,NYM1*0.0>
C     BZFRAC(1-NZ)<1.0,NZM1*0.0>
C     SERVICE SUBROUTINE FOR SUB-DOMAIN SPECIFICATION (FOR IGEN=1
C     ONLY):
C        CALL DOMAIN(ID,IXF,IXL,IYF,IYL,IZF,IZL)
C     XE(1-NYP1,1-NZP1,1-ND)<(NYP1*NZP1*ND)*1.0>,
C     XW(1-NYP1,1-NZP1,1-ND),
C     YN(1-NXP1,1-NZP1,1-ND)<(NXP1*NZP1*ND)*1.0>,
C     YS(1-NXP1,1-NZP1,1-ND),
C     ZH(1-NXP1,1-NYP1,1-ND)<(NXP1*NYP1*ND)*1.0>,
C     ZL(1-NXP1,1-NYP1,1-ND),STORSA(1-6)<6*.F.>,STORWD(1-6)<6*.F.>,
C     STORP<.F.>,STORPE<.F.>,STORPN<.F.>,STORPH<.F.>,STOUNV<.F.>,
C     PRTBFC<.F.>,DARCY,BFPLOT<.F.>
C     CYCLIC BOUNDARY CONDITIONS ARE DEFAULTED INACTIVE ;
C     TO ACTIVATE THEM AT SELECTED IZ SLABS USE SERVICE SUBROUTINE:
C        CALL XCYIZ(IZ,.TRUE.)
C     SERVICE SUBROUTINE TO DEACTIVATE CURVATURE TERMS IN U, V
C     AND W EQUATIONS ASSOCIATED WITH CURVATURE OF IX, IY, IZ
C     GRID LINES RESPECTIVELY:
C        CALL UCURVE(IZ,.FALSE.)
C        CALL VCURVE(IZ,.FALSE.)
C        CALL WCURVE(IZ,.FALSE.)
C     NCART<1>
C    *WARNINGS||||||
C     -------------
C       A) WHEN USING BFCS STOVAR(H3), STOVAR(C4), STOVAR(21) ARE
C          AVAILABLE ONLY FOR STORING NON-ORTHOGONAL VELOCITY
C          COMPONENTS.
C       B) MULTI-RUNS ARE NOT ALLOWED WITH BFC OPTION.
C       C) MOVING GRID,TWO-PHASE AND PARABOLIC OPTIONS ARE NOT
C          AVAILABLE WITH BFC OPTION.
C       D) KE-EP TURBULENCE MODEL SHOULD BE USED WITH BFC'S ONLY
C          WHEN THE MAIN FLOW IS IN THE IZ DIRECTION.
C       E) BUILT-IN GRAVITY TERMS DO NOT TAKE ACCOUNT OF BFC'S.
C    *NOTES
C     -----
C       A) THE STANDARD VELOCITY-FIELD PRINTOUT FOR THE
C          VELOCITY RESOLUTES IS ACTIVATED IN THE USUAL
C          WAY. AN ADDITIONAL OPTION EXISTS FOR PRINTING THE
C          CARTESIAN VELOCITY-COMPONENTS WHICH MAY BE
C          ACTIVATED BY SETTING THE FOLLOWING LOGICALS:
C               STOVAR(U2)=.T. FOR U-COMPONENT (CARTESIAN)
C               STOVAR(V2)=.T. FOR V-COMPONENT (CARTESIAN)
C               STOVAR(W2)=.T. FOR W-COMPONENT (CARTESIAN)
C          SIMILARLY PRINTOUT OF NON-ORTHOGONAL VELOCITY
C          COMPONENTS MAY BE ACTIVATED AS FOLLOWS:
C               STOVAR(C4)=.T. FOR U-COMPONENT (NON-ORTHOG)
C               STOVAR(H3)=.T. FOR V-COMPONENT (NON-ORTHOG)
C               STOVAR(21)=.T. FOR W-COMPONENT (NON-ORTHOG)
C       B) BFC (TO ACTIVATE THE BFC OPTION), IGEN (THE CODE FOR METHOD
C          OF GRID SPECIFICATION), ND (NUMBER OF SUB-DOMAINS) AND
C          NBFC (THE F1 ARRAY DIMENSION), MUST BE SET BEFORE
C          "STANDARD BFC SECTION 2".                ======
```

```
C          ALL OTHER BFC DATA MUST BE SET AFTER "STANDARD BFC
C          SECTION 2.                           =====
C       C) NXP1, NYP1, NZP1 STORE NX+1, NY+1, NZ+1; THESE ARE
C          AVAILABLE TO USER AFTER STANDARD BFC SECTION 2.
C       D) FOR IGEN=1 USE BXFRAC,BYFRAC & BZFRAC IN PLACE OF
C          XFRAC,YFRAC & ZFRAC.
C-------------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 STARTS:
C     DEFAULT SETTINGS:
      NCART=10
      BFC=.TRUE.
      IGEN=1
      ND=1
      NBFC=5000
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 1 ENDS.:
C    *USER SETS BFC, IGEN, ND AND NBFC HERE:
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 STARTS:
      CALL SB4I(NXP1,NX+1,NYP1,NY+1,NZP1,NZ+1,I,0)
      IF(BFC) CALL BFCDFT(NBFC,XE,XW,YN,YS,ZH,ZL,ND,NXP1,NYP1,
     &                    NZP1,NZ)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD BFC SECTION 2 ENDS.
C    *USER SETS ALL OTHER BFC VARIABLES HERE:
C    *USING NONIFORM GRID 1-8
      GTH=65.E-3
      GTL=150.E-3
      GBETA=4.
      GBETA=GBETA*3.1415927/180
      GTAB=TAN(GBETA)
      DELMAX=2.E-3
      GNBL=5.
      GPWR=4.
      DO 64 IY=1,5
   64 BYFRAC(IY)=(FLOAT(IY)/GNBL)**GPWR*DELMAX/GTH
      BYFRAC(6)=BYFRAC(5)+3.E-3/GTH
      DEL=(1.-BYFRAC(6))/(FLOAT(NY)-GNBL-1)
      DO 65 IY=7,NY
   65 BYFRAC(IY)=BYFRAC(IY-1)+DEL
C----------ZZ-----------
      BZFRAC(1)=10.E-3
      DO 66 IZ=2,5
   66 BZFRAC(IZ)=10.E-3+BZFRAC(IZ-1)
      BZFRAC(6)=BZFRAC(5)+5.E-3
      DO 67 IZ=7,9
   67 BZFRAC(IZ)=BZFRAC(IZ-1)+2.E-3
      DO 68 IZ=10,11
   68 BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
      BZFRAC(12)=BZFRAC(11)+1.E-3
      DO 77 IZ=13,14
   77 BZFRAC(IZ)=BZFRAC(IZ-1)+.5E-3
      DO 78 IZ=15,15
   78 BZFRAC(IZ)=BZFRAC(IZ-1)+1.E-3
      BZFRAC(16)=BZFRAC(15)+1.E-3
      BZFRAC(17)=BZFRAC(16)+2.E-3
      BZFRAC(18)=BZFRAC(17)+7.E-3
      DO 69 IZ=19,22
   69 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
      BZFRAC(23)=BZFRAC(22)+3.E-3
      BZFRAC(24)=BZFRAC(23)+2.E-3
      BZFRAC(25)=BZFRAC(24)+2.E-3
      BZFRAC(26)=BZFRAC(25)+3.E-3
      BZFRAC(27)=BZFRAC(26)+5.E-3
      DO 71 IZ=28,NZ
   71 BZFRAC(IZ)=BZFRAC(IZ-1)+10.E-3
      DO 72 IZ=1,NZ
   72 BZFRAC(IZ)=BZFRAC(IZ)/GTL
      CALL DOMAIN(1,1,NX,1,NY,1,NZ)
      DO 61 IX=1,NXP1
      DO 62 IY=1,NYP1
      ZL(IX,IY,1)=0.0
   62 ZH(IX,IY,1)=GTL
      DO 63 IZ=1,NZP1
```

```
      YN(IX,IZ,1)=GTH
   63 YS(IX,IZ,1)=0.0
C  YS(IX,13,1) SHOULD COME AFTER
      DO 662 IZ=5,25
CBL   DO 662 IZ=16,25
  662 YS(IX,IZ,1)=(BZFRAC(IZ-1)-BZFRAC(3))*GTAB*GTL
CBL   DO 663 IZ=13,15
CBL   GZ12=(BZFRAC(IZ-1)-BZFRAC(11))*GTL-.5E-3
CBL663YS(IX,IZ,1)=SQRT(YS(IX,16,1)*GZ12*2.-GZ12**2)
      DO 664 IZ=26,NZ
  664 YS(IX,IZ,1)=YS(IX,25,1)
   61 CONTINUE
      STORSA(IFIX(LOW))=.TRUE.
      STORSA(IFIX(HIGH))=.TRUE.
      STORSA(IFIX(SOUTH))=.TRUE.
      STORWD(IFIX(SOUTH))=.TRUE.
      STORP=.TRUE.
      PRTBFC=.TRUE.
CDAR  DARCY=1.E10
C-----------------   --------------   -----------------
C---  GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
C    *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
C     CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
C     IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION; 'TYPE'= EAST,
C     WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
C     OVER REGION IXF,...IZL.
C    *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
C     PC(NX,NY,NZ) ABOVE.
C    *FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
C     THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
C     AUTOMATICALLY ZEROED.
C    *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
C     XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
C    *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
C     BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
C     IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
C     IN THIS CASE, THE USER  M U S T   SET  A L L  ELEMENTS OF
C     ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
C     CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
C     ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
C     'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
C    *CONPOR  M U S T  N O T  BE USED IN CONJUNCTION WITH EXPLICIT
C     SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
C-----------------   --------------   -----------------
C---  GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
C     SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
C     USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
C     P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
      SOLVAR(P1)=.TRUE.
      SOLVAR(PP)=.TRUE.
      SOLVAR(V1)=.TRUE.
      SOLVAR(W1)=.TRUE.
      SOLVAR(H1)=.TRUE.
CT    SOLVAR(KE)=.TRUE.
CT    SOLVAR(EP)=.TRUE.
      STOVAR(V2)=.TRUE.
      STOVAR(W2)=.TRUE.
      STOVAR(C1)=.TRUE.
      STOVAR(C2)=.TRUE.
      STOVAR(C3)=.TRUE.
C-----------------   --------------   -----------------
C---  GROUP 9. VARIABLE LABELS :
C     TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
C     2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
C     2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H****>
      TITLE(C1)=TITC1
      TITLE(C2)=TITC2
      TITLE(C3)=TITC3
      TITLE(PP)=TITPP
C-----------------   --------------   -----------------
C---  GROUP 10 PROPERTIES:
C     IRHO1<1>,IRHO2<1>,RHO1<1.0>,RHO2<1.0>,
```

```
C       ARHO1<1.0>,BRHO1<1.0>,CRHO1<1.0>
C       IEMU1<1>,EMU1<1.0>,EMULAM<1.E-10>
C       IHSAT,H1SAT,H2SAT,PSATEX<1.0>
C       SIGMA(1-25)<1.0,2.0,1.,1.E10,1.,1.E10,1.,1.E10,
C       4*1.0,1.314,1.0,1.E10,10*1.0>
        IRHO1=-1
        PTOT=55.E5
        TOT=555.55
        RAIR=287.
        GAMA=1.35
        CP=RAIR/(1-1/GAMA)
        TW=323.
        HWALL=TW*CP
        HTOT=CP*TOT
        RHTOT=PTOT/TOT/RAIR
        LOGIC(87)=.TRUE.
        ARHO1=RHTOT/PTOT**(1/GAMA)
        BRHO1=1./GAMA
C  TURBULENT OR LAMINAR
        IEMU1=-1
C       IEMU1=1
        JEMU1=IEMU1
        EMU1=1.E-5
        EMULAM=EMU1
        GEMU1=EMU1
        GPR=.7
        SIGMA(24)=GPR
        SIGMA(14)=GPR
C------------------    ---------------    -------------------
C---    GROUP 11 INTER-PHASE TRANSFER PROCESSES :
C       ICFIP,CFIPS,IMDOT,CMDOT,CA1I<1.E6>,CA2I<1.E6>
C------------------    ---------------    -------------------
C---    GROUP 12 SPECIAL SOURCES :
C       ISPCSO(1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF
C------------------    ---------------    -------------------
C---    GROUP 13 INITIAL FIELDS :
C       FIINIT(1-25)<25*1.E-10>
C     MACH NO. OF FREE STREAM
        GMACH=3.2
        A=1+(GAMA-1)/2*GMACH**2
        TE=TOT/A
        RHE=RHTOT/A**(1/(GAMA-1))
        PSTAT=PTOT/A**(GAMA/(GAMA-1))
        RHO1=ARHO1*PSTAT**BRHO1
        SONIC=SQRT(GAMA*RAIR*TE)
        WIN=SONIC*GMACH
        RKEIN=0.01*WIN**2
        EPIN=0.16*RKEIN**1.5/GTH/2.
        FIINIT(W1)=WIN
        FIINIT(P1)=PSTAT
        FIINIT(H1)=HTOT
        FIINIT(KE)=RKEIN
        FIINIT(EP)=EPIN
C------------------    ---------------    -------------------
C---    GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C       ILOOP1,ILOOPN,XCYCLE<.F.>,PBAR,REGION(1-10)<10*.T.>
C     *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE..  THE USER SHOULD
C       SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
        DO 14 I=1,10
   14   REGION(I)=.FALSE.
C------------------    ---------------    -------------------
C---    GROUP 15 TO 24; REGIONS 1 TO 10
C---    ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE
C       USER, PREFERABLY BY WAY OF:-
C       CALL PLACE(IREGN,TYPE,IXF,IXL,IYF,IYL,IZF,IZL) &
C       CALL COVAL(IREGN,VARBLE,COEFF,VALUE)
        CALL PLACE(1,LOW,1,NX,1,NY,1,1)
C       CALL COVAL(1,M1,FIXFLU,WIN*RHE)
CDAR    CALL COVAL(1,M1,1.E-20,1.E+20*WIN*RHE)
        GCM=2*GAMA/WIN/(GAMA-1)
        GVM=PTOT*RHE/RHTOT
        CALL COVAL(1,M1,GCM,GVM)
```

```
      CALL COVAL(1,W1,ONLYMS,WIN)
      CALL COVAL(1,H1,ONLYMS,HTOT)
C     CALL COVAL(1,KE,ONLYMS,RKEIN)
C     CALL COVAL(1,EP,ONLYMS,EPIN)
      CALL PLACE(2,HIGH,1,NX,1,NY,NZ,NZ)
C     CALL COVAL(2,M1,FIXVAL,PSTAT*0.)
      CALL COVAL(2,M1,1000*WIN*RHE/PSTAT,PSTAT)
      CALL COVAL(2,H1,ONLYMS,HTOT)
C   WALL ALONG THE VANE  IZ(11,NZ)
      GCM=EMU1/(.5*BYFRAC(1)*GTH)
      DY1=BYFRAC(1)*GTH
      GOEFF=EMU1/(0.5*DY1)
      GOEFH=EMU1/(0.5*DY1*SIGMA(24))
      CALL PLACE(3,SOUTH,1,NX,1,1,4,NZ)
C     CALL COVAL(3,W1,GOEFF,0.)
C     CALL COVAL(3,H1,GOEFH,HWALL)
      CALL COVAL(3,W1,WALL,0.)
      CALL COVAL(3,H1,WALL,HWALL)
CT    CALL COVAL(3,KE,WALL,0.)
CT    CALL COVAL(3,EP,WALL,0.)
C-------------------    ---------------    -------------------
C---  GROUP 25 GROUND STATION :
C     GROSTA<.F.>,NAMLST<.F.>
C    *NAMLST ACTIVATES NAMELIST IN GROUND.
      GROSTA=.TRUE.
C-------------------    ---------------    -------------------
C---  GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C     WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
      WHOLEP=.TRUE.
C-------------------    ---------------    -------------------
C---  GROUP 27 SWEEP AND ITERATION NUMBERS :
C     FSWEEP<1>,LSWEEP<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
C     LITER(1-25)<9*1,-1,15*1>
C     IVELF<1>,NVEL<1>,IVELL<10000>,
C     IKEF<1>,NKE<1>,IKEL<10000>,
C     IENTF<1>,NENT<1>,IENTL<10000>,
C     ICNCF<1>,NCNC<1>,ICNCL<10000>,
C     IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
C     IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
      LSWEEP=400
      GSWP=LSWEEP
CR    FSWEEP=200
      LITER(PP)=20
      LITER(V1)=5
      LITER(W1)=5
C     LITHYD=2
C-------------------    ---------------    -------------------
C---  GROUP 28 TERMINATION CRITERIA :
C     ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
      ENDIT(1)=1.E-5
C-------------------    ---------------    -------------------
C---  GROUP 29 RELAXATION  :
C     RLXP<1.>,RLXPXY<1.>,RLXPZ<1.>,RLXRHO<1.>,RLXMDT<1.>,
C     DTFALS(3-25)<23*1.E10>
      DTFALS(W1)=1.E-5
      DTFALS(V1)=1.E-5
      RLXP=.2
C-------------------    ---------------    -------------------
C---  GROUP 30 LIMITS :
C     VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
C     TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
C     EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
C-------------------    ---------------    -------------------
C---  GROUP 31 SLOWING DEVICES : SLORHO<1.>,SLOEMU<1.>
      SLORHO=.2
C-------------------    ---------------    -------------------
C---  GROUP 32 PRINT-OUT OF VARIABLES :
C     PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
      PRINT(C1)=.TRUE.
      PRINT(C2)=.TRUE.
      PRINT(C3)=.TRUE.
      PRINT(PP)=.TRUE.
```

```
C------------------   ---------------   -------------------
C---   GROUP 33 MONITOR PRINT-OUT :
C      IXMON<1>,IYMON<1>,IZMON<1>,NPRMON<1>,NPRMNT<1>
       NPRMON=5
       IYMON=2
       IZMON=12
C------------------   ---------------   -------------------
C---   GROUP 34 FIELD PRINT-OUT CONTROL :
C      NPRINT<100>,NTPRIN<100>,NXPRIN<1>,NYPRIN<1>,NZPRIN<1>,
C      IZPRF<1>,ISTPRF<1>,IZPRL<10000>,ISTPRL<10000>
C      NUMCLS<10>,KOUTPT
       NPRINT=LSWEEP
C------------------   ---------------   -------------------
C---   GROUP 35 TABLE CONTROL :
C      TABLES<.F.>,NTABLE,NTABVR,LINTAB,NPRTAB,NMON,
C      ITAB(1-8),MTABVR(1-8)
C------------------   ---------------   -------------------
C      GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C      MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C---   GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C      IZPR1<1>,IZPR2<1>,ISTPR1<1>,ISTPR2<1>
C------------------   ---------------   -------------------
C---   GROUP 37 DEBUG SWEEP AND SUBROUTINES :
C      KEMU,KMAIN,KINDEX,KGEOM,KINPUT,KSODAT,KCOMPF,KSORCE,
C      KSOLV1,KSOLV2,KSOLV3,KCOMPP,KADJST,KFLUX,KSHIFT,KDIF,
C      KCOMPU,KCOMPV,KCOMPW,KCOMPR,KWALL,KDBRHO<-1>,KDBEXP,KDBMDT
C      KDBGEN
C------------------   ---------------   -------------------
C---   GROUP 38 MONITOR,TEST,AND FLAG :
C      MONITR<.F.>,FLAG<.F.>,TEST<.T.>,KFLAG<1>
C      END OF MAINTENANCE-ONLY SECTION
C------------------   ---------------   -------------------
C---   GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C      IERRP<1000>,RESREF(1,3-24)<25*1.>,RESMAP<.F.>,
C      RESID(1-25)<2*.F.,23*.T.>,KOUTPT
       RESREF(1)=WIN*RHE
       RESREF(7)=WIN*RESREF(1)
       RESREF(5)=WIN*RESREF(1)*0.1
       RESREF(H1)=HTOT*RESREF(1)
       RESREF(KE)=RKEIN*RESREF(1)
       RESREF(EP)=EPIN*RESREF(1)
       IERRP=LSWEEP/10
       KOUTPT=LSWEEP/10
C------------------   ---------------   -------------------
C---   GROUP 40 SPECIAL DATA : LOGIC(1..10),INTGR(1..10),RE(21..30),
C      NLSP<1>,NISP<1>,NRSP<1>,SPDATA<.F.>,LSPDA(1),ISPDA(1),RSPDA(1)
C      USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C      TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C      SATELLITE TO GROUND, BUT IF REQUIREMENTS EXCEED THIS
C      PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C      ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C-----------------------------------------------------------
C---   GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>,RESTRT<.F.>,KINPUT
       SAVEM=.TRUE.
       BFPLOT=.TRUE.
C      RESTRT=.TRUE.
C------------------   ---------------   -------------------
C------------------   ---------------   -------------------
C---   GROUP 43 GRAFFIC :
C      GRAPHS<.F.>,ORTHOG<.T.>,ANTSYM,NPRT<1>,ITITL<5*4H****>
C---   FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C      PHI1(NX*NY*NZ*NM)
C      PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)) , WHERE
C      NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C      IBLK=0 IF BLOCK=.FALSE.,=4 IF A 3D RUN,
C      =3 IF A 2D.YZ RUN.
C------------------   ---------------   -------------------
          IF(IRUN.EQ.1) GO TO 900
  900  CONTINUE
C---   ALL RUNS
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 4 STARTS:
```

```
C-------------------------------------------------------------
C  WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...
           IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
           IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)
       IF(BFC) CALL WRTBFC(14,NBFC,XE,XW,YN,YS,ZH,ZL,
     &ND,NX+1,NY+1,NZ+1,NZ,PRTBFC)
C      OLD PRACTICES RETAINED FOR REFERENCE:
C          IF(SPDATA) CALL SPCDAT(IRUN)
C          IF(BLOCK) CALL PORDAT(IRUN)
           IF(GRAPHS) CALL SORT(IRUN)
           IF(RESTRT) GO TO 902
         DO 901 INDVAR=1,25
           IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901
      CALL FLDDAT(IRUN)
      GO TO 902
  901 CONTINUE
  902 CALL DATAIO(WRT,10)
           IF(MONITR) CALL DATAIO(WRT,-6)
 999  CONTINUE
      STOP
      END
C***   IGEN=1 SO BFCXYZ NOT REQUIRED.
C***   COMMENT OUT BOTH VERSIONS.
C--------------------------------------------------------------
      SUBROUTINE BFCXYZ (NXP1,NYP1,NZP1)
      RETURN
      END
```

Appendix B

Ground Listing

```
C$DIRECTIVE**MAIN    AMI LEITNER
C      LECGRD   LAST GEO. NZ=27 NY=18 LAMINAR FLOW
C     *FILE NAME: MODBFCGD.FTN
C     *INCLUDE DED SUBROUTINES: THE MODELS OF MAIN, GROUND & STRIDE.
C     *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983)
C      WITH BODY-FITTED COORDINATES INSTRUCTION SUPPLEMENT
C      (SUMMER 1984).
C     *SATELLITE FILE NAME: MODSTL.FTN
      COMMON/ISHIFT/III(57),NFMAX
C SET F-ARRAY DIMENSION AS NEEDED, & SET NFMAX ACCORDINGLY.
C     FOR BFC'S ALSO SET F1-ARRAY DIMENSION AS NEEDED ,AND SET
C     NF1MAX ACCORDINGLY.
      COMMON/F0B/F1(10000)
      COMMON/NF0B/NF1MAX
      COMMON F(25000)
      NFMAX=25000
      NF1MAX=10000
      CALL MAIN1
      STOP
      END
C$DIRECTIVE**GROUND
      SUBROUTINE GROUND(IRN,ICHAP,ISTP,ISWP,IZED,INDVAR)
      INCLUDE (CMNGUS)
      INCLUDE (GUSSEQ)
C      INCLUDE NMLIST
      LOGICAL BFC
      EQUIVALENCE (LOGIC(20),BFC)
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C---------------------------------------------------------------------
C+++++MEANING OF SUBROUTINE ARGUMENTS:
C     IRN=RUN NUMBER; ICHAP=CHAPTER CALLED; ISTP=TIME STEP;
C     ISWP=SOLUTION SWEEP; IZED=Z-SLAB; INDVAR: SEE CHAPTERS BELOW.
C+++++USER-INTRODUCED VARIABLES & ARRAYS:
C     TO AVOID CONFLICT WITH VARIABLE NAMES USED IN COMMON, ALL
C     VARIABLES INTRODUCED BY THE USER SHOULD HAVE NAMES STARTING
C     WITH 'G' IF REAL, 'J' IF INTEGER, AND 'G' OR 'J' IF LOGICAL.
C     THUS GDZ(IZ) MIGHT BE A Z-INTERVAL ARRAY;
C     GW1(IY,IX) A 2-D ARRAY FOR AXIAL VELOCITY; ETC.
C     USER-GENERATED SUBROUTINES SHOULD BE NAMED CORRESPONDINGLY,EG
C     SUBROUTINE GVISC(GTEMP,GCNC,GVSC), FOR COMPUTING VISCOSITY
C     FROM CONCENTRATION & TEMPERATURE.
C+++++GROUND-TO-EARTH CONNECTING SUBROUTINES:
C    *USE GET(NAME,GARRAY,NY,NX) TO PUT VALUES OF VARIABLE NAMED
C     'NAME' INTO ARRAY 'GARRAY' DIMENSIONED GARRAY(NY,NX).
C    *USE SET(NAME,IXF,IXL,IYF,IYL,GARRAY,NY,NX) TO SET VARIABLE
C     'NAME' TO GARRAY(IY,IX) OVER THE REGION: IXF-IXL & IYF-IYL.
C    *USE PRNSLB(NAME) TO PRINT VARIABLE 'NAME' OVER X-Y PLANE.
C    *USE ADD(NAME,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C     TO ADD SOURCE TO VARIABLE NAMED 'NAME' (SEE CHAPTER 5).
C    *USE READIZ(IZED) IN CHAPTERS 1, 2, 8, & 9 TO ACCESS P1,...DM
C     & VOL,...AHDZ. (SEE FOOTNOTE TO LEGALITY TABLE)
C    *USE GET1D(NAME,GARRAY,NDIM) TO PUT VARIABLE NAMED 'NAME' IN
C     ONE-D ARRAY 'GARRAY' DIMENSIONED NDIM, THUS:
C     CALL GET1D(NAME,GNX,NX) FOR XG,...DXG & DIMENSION GNX(NX);
C     CALL GET1D(NAME,GNY,NY) FOR YG,...RV & DIMENSION GNY(NY);
C     CALL GET1D(NAME,GNZ,NZ) FOR ZG,...WGRID & DIMENSION GNZ(NZ).
C+++++LEGALITY TABLE FOR USE OF EARTH-CONNECTING SUBROUTINES:
C     ENTRIES IN TABLE GIVE CHAPTERS IN WHICH SUBROUTINES CAN BE
C     USED FOR VARIABLES IN LEFT-HAND COLUMN. (SUBROUTINE
C     STRIDE IS REGARDED AS BEING IN CHAPTER 3)
C     -----------------------------------------------------------
C     : VARIABLE:: GET &   :   SET  : ADD    : READIZ : GET1D :
C     :          ::  PRNSLB :        :        :        :       :
C     -----------------------------------------------------------
C     :P1 - RZ   ::    ALL  :  6 & 7 :   5    : 1,2,8,9:  NONE :
C     :P10 - RZH::3-7, 10-16:    3   :  NONE  :  NONE  :  NONE :
C     :VOL -AHDZ::    ALL  :    3   :  NONE  : 1,2,8,9:  NONE :
C     :D1DP      ::    NONE :   10   :  NONE  :  NONE  :  NONE :
C     :D2DP      ::    NONE :   11   :  NONE  :  NONE  :  NONE :
C     :MU1,MU1H  :: 5,13-16 :   12   :  NONE  :  NONE  :  NONE :
C     :EXCO(L,H)::    NONE :   13   :  NONE  :  NONE  :  NONE :
C     :CFP       ::     5   :   14   :  NONE  :  NONE  :  NONE :
```

FILE: VANTGRD FORTRAN A1

```
C      :MDT       ::      5     :    15    :  NONE   :  NONE   :  NONE :
C      :HST1,HST2::  5 & 15   :    16    :  NONE   :  NONE   :  NONE :
C      :XG -WGRID::    NONE   :   NONE   :  NONE   :  NONE   :  ALL  :
C      --------------------------------------------------------------
C      NOTES ON ABOVE TABLE:
C     *IN CHAPTERS 1, 2, 8, & 9 VARIABLES P1...DM & GEOMETRY
C      VOL...AHDZ CAN BE ACCESSED BUT ONLY IN CONJUNCTION WITH
C      USE OF READIZ, THUS:
C      DO 1 IZED=1,NZ
C      CALL READIZ(IZED)
C    1 CALL GET(... AS REQUIRED..)
C     *GEOMETRY ACCESSED BY READIZ IS THAT AT INITIAL TIME.
C     *D1DP & D2DP ONLY ACCESSIBLE IN UNSTEADY FLOWS.
C+++++GROUND SERVICE SUBROUTINES:
C     *USE CONTUR(NAME,IPLANE,ILOC,NINT,I1,I2,J1,J2,GARRAY,NDIM) FOR
C      LINE-PRINTER PLOTS OF CONTOURS. 'NAME' = U1,...C4;
C      'IPLANE'= XPLANE, YPLANE, OR ZPLANE; ILOC SETS IX, IY, OR
C      IZ LOCATION OF IPLANE; I1, I2, J1, & J2 SET FIRST & LAST
C      CELLS IN HORIZ. & VERT. ON PLOT; GARRAY IS 1-D WORKING ARRAY
C      OF DIMENSION NX*NY, NX*NZ, OR NY*NZ DICTATED BY IPLANE; &
C      NDIM SETS VALUE OF DIMENSION OF GARRAY.
C     *USE FLD2DA(TITLE,GARRAY,NY,NX) TO PRINT ANY ARRAY DIMENSIONED
C      GARRAY(NY,NX); SET 'TITLE' TO REQUIRED NAME ( 4 HOLLERITH
C      CHARACTERS ONLY).
C     *USE FLD3DA(TITLE,GARRAY,NX,NY,NZ,IPLANE,ILOC) TO PRINT ANY
C      ARRAY DIMENSIONED GARRAY(NX,NY,NZ) IN PLANE SPECIFIED BY
C      'IPLANE' & 'ILOC' AS FOR CONTUR ABOVE; SET 'TITLE' AS FOR
C      FLD2DA.
C      VARIABLE NAMES FOR USE IN GROUND:
       COMMON/TYPE/CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME,WALL
       COMMON/VAR/P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
      &KE,EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
       COMMON/VAROLD/P1O,PPO,U1O,U2O,V1O,V2O,W1O,W2O,R1O,R2O,RSO,
      &KEO,EPO,H1O,H2O,H3O,C1O,C2O,C3O,C4O,RXO,RYO,RZO,S1O,S2O
       COMMON/VARLOW/P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
      &KEL,EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
       COMMON/VARHI/P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
      &KEH,EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
       COMMON/GMTRY/VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
       COMMON/PROP/D1,D2,D1DP,D2DP,MU1,MU1LAM,EXCO,CFP,MDT,HST1,HST2
       COMMON/PRPOLD/D1O,D2O
       COMMON/PRPLOW/D1L,D2L,EXCOL
       COMMON/PRPHI/D1H,D2H,MU1H,EXCOH
       COMMON/VARNX/XG,XU,DXU,DXG
       COMMON/VARNY/YG,YV,DYV,DYG,R,RV
       COMMON/VARNZ/ZG,ZW1,DZW,DZG,WGRID
       COMMON/GDMSCI/XPLANE,YPLANE,ZPLANE,ITNO
       COMMON/GDMSCL/LSLAB,MSLAB,HSLAB,LAMMU
       REAL NORTH,LOW
       INTEGER    P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
      &EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
       INTEGER    P1O,PPO,U1O,U2O,V1O,V2O,W1O,W2O,R1O,R2O,RSO,
      &EPO,H1O,H2O,H3O,C1O,C2O,C3O,C4O,RXO,RYO,RZO,S1O,S2O
       INTEGER    P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
      &EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
       INTEGER    P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
      &EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
       INTEGER VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
       INTEGER D1,D1DP,D2,D2DP,EXCO,CFP,HST1,HST2
       INTEGER D1O,D2O,D1L,D2L,EXCOL,D1H,D2H,EXCOH
       INTEGER XG,XU,DXU,DXG,YG,YV,DYV,DYG,R,RV,ZG,ZW1,DZW,
      &DZG,WGRID
       INTEGER XPLANE,YPLANE,ZPLANE
       LOGICAL LSLAB,MSLAB,HSLAB,LAMMU,LSPDA
       EQUIVALENCE (M1,R1),(M2,R2)
C      SATLIT-EQUIVALENT IRUN:
       EQUIVALENCE (IRUN,INTGR(11))
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C      ARRAYS ( DIMENSIONED NY,NX ) FOR USE WITH 'ADD':
       DIMENSION CVAR(1,1),VVAR(1,1),CM(1,1),VM(1,1),ZERO(1,1)
       DIMENSION GP(30,1),GH(30,1),GD(30,1),GV(30,1),GW(30,1)
```

```
     1 ,GMACH(30,1),GTEMP(30,1),GVISC(30,1),GWH(30,1),GWM(30,1)
     2 ,GKE(30,1),GC3(30,1),GYG(30,1),GXX(30,1),GYY(30,1),GZZ(30,1)
C     SPECIAL-DATA ARRAYS DIMENSIONED & DIMENSION VALUES SET HERE:
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(1)
C     USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      EQUIVALENCE (RAIR,RE(21)),(GAMA,RE(22)),(GSWP,RE(23)),
     1(GPR,RE(24)),(GTW,RE(25)),(GEMU1,RE(26)),(JEMU1,INTGR(1))
      DATA NLSP,NISP,NRSP/1,1,1/
      DATA CVAR,VVAR,CM,VM,ZERO/5*0.0/
C     USER PLACES HIS DATA STATEMENTS HERE.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
C     PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C     STATEMENTS OF THIS SECTION.
      DATA NUMCH4 / 0 /
          IF(SPDATA)
     &CALL RDSPC(IRN,INTGR(12),LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
      CALL GRDUTY(IRN,ICHAP,IZED,INDVAR)
      IF(BFC) CALL BFCGRD(IRN,ICHAP,ISWP,IZED,INDVAR)
          IF(ICHAP.EQ.-5) GO TO 10
          IF(ICHAP.LE.0.OR.ICHAP.GT.16) RETURN
      GO TO (100,200,300,4999,500,600,700,800,900,1000,1100,1200,
     &1300,1400,1500,1600),ICHAP
      RETURN
 4999 NUMCH4= NUMCH4 + 1
      IF (MOD(NUMCH4,2).EQ.1) GO TO 400
      RETURN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C-----------------------------------------------------------------
C     CHAPTER 0: MODIFY SATLIT DATA, AT START OF EACH IRN.
C-----------------------------------------------------------------
   10 CONTINUE
C         IF(.NOT.NAMLST) RETURN
C         IF(IRN.EQ.NRUN) DATFIL=.FALSE.
C---  READ SATLIT DATA NAMELIST HERE
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 1 TO 24  )
C     READ(20,G1G24)
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 25 TO 42 )
C     READ(20,G25G42)
      RETURN
C------------------------------------------------------------------
C     CHAPTER 1: CALLED AT THE START OF EACH TIME STEP.
C     SET 'DT' HERE WHEN TLAST SET NEGATIVE IN BLOCK DATA.
C     'ATIME + DT' GIVES THE END TIME OF THE CURRENT TIME STEP.
C     NOT ACCESSED IF STEADY,OR PARABOLIC.
C------------------------------------------------------------------
  100 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 2: CALLED AT THE START OF EACH SWEEP.
C------------------------------------------------------------------
  200 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 3: CALLED AT THE START OF EACH SLAB;
C     NOT ACCESSED IF PARABOLIC, BUT 'STRIDE' IS.
C------------------------------------------------------------------
  300 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 4: CALLED AT THE START OF EACH RE-CALCULATION OF
C     VARIABLES P1,...C4 AT CURRENT SLAB. ITNO= ITERATION NUMBER.
C------------------------------------------------------------------
  400 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 5: GROUND CALLED WHEN SOURCE TERM IS COMPUTED.
C     INDVAR GIVES DEPENDENT VARIABLE IN QUESTION IE. U1,...C4.
C     TO ADD SOURCE TO DEPENDENT VARIABLE C1(SAY) FOR IX=IXF,IXL
C     AND IY=IYF,IYL INSERT STATEMENT:
C     IF(INDVAR.EQ.C1)
```

```
C     &CALL ADD(INDVAR,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C      NOTES ON 'ADD':
C     *SOURCE= (CVAR(IY,IX)+AMAX1(0.0,MASFLO))*(VVAR(IY,IX)-PHI),
C      WHERE 'PHI' IS IN-CELL VALUE OF VARIABLE IN QUESTION.
C     *'MASFLO'= CM(IY,IX)*(VM(IY,IX)-P),
C      WHERE 'P' IS THE IN-CELL PRESSURE.
C     *FOR INDVAR= M1, OR =M2, SOURCE ADDED IS 'MASFLO' ONLY,
C      EXCEPT FOR ONEPHS=.F. & MASFLO < 0.0 (IE. OUTFLOW) WHEN
C      CM(IY,IX) IS MULTIPLIED BY R1*D1 (FOR M1) & R2*D2 (FOR M2).
C     *BOTH 'CVAR' & 'CM' ARE MUTLIPLIED BY CELL-GEOMETRY QUANTITY
C      DICTATED BY SETTING OF 'TYPE' (=CELL, EAST AREA,..VOLUME).
C     *TYPE-SPECIFIED AREAS ARE CALCULATED AS IF BLOCKAGE ABSENT,
C      BUT 'VOLUME' WITH ACCOUNT FOR ITS PRESENCE.
C     *FOR ALL SOLVED VARIABLES, INCLUDE DING M1 ( & M2 WHEN ONEPHS=F),
C      IF 'CM'> 0.0 CALL 'ADD'; FOR M1 & M2 ALTHOUGH 'CVAR' & 'VVAR'
C      HAVE NO SIGNIFICANCE THEY MUST BE ENTERED AS ARGUMENTS.
C     *'CVAR', 'VVAR', 'CM' & 'VM' MUST BE DIMENSIONED NY,NX.
C-----------------------------------------------------------------------
  500 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 6: CALLED AT THE END OF EACH VARIABLE-RECALCULATION
C     CYCLE COMMENCED AT CHAPTER 4. ITNO = ITERATION NUMBER.
C-----------------------------------------------------------------------
  600 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 7: CALLED AT END OF EACH SLAB-WISE CALCULATION.
C-----------------------------------------------------------------------
  700 CONTINUE
      IF(FLOAT(ISWP).LT.GSWP) RETURN
      CALL GET(P1,GP,NY,NX)
      CALL GET(H1,GH,NY,NX)
      CALL GET(D1,GD,NY,NX)
      CALL GET(V1,GV,NY,NX)
      CALL GET(W1,GW,NY,NX)
      CALL GET(KE,GKE,NY,NX)
C     CALL GET1D(YG,GYG,NY)
      CALL GRED1(39,IZED,GYG,NY,NX)
      CALL GRED3(57,IZED,GXX,GYY,GZZ,NY,NX)
      GCP=RAIR/(1.-1/GAMA)
      DO 701 I=1,NY
      GSON=SQRT(GAMA*GP(I,1)/GD(I,1))
      GAV=SQRT(GV(I,1)**2+GW(I,1)**2)
      GMACH(I,1)=GAV/GSON
  701 GTEMP(I,1)=GP(I,1)/GD(I,1)/RAIR
C 701 GTEMP(I;1)=(GH(I,1)-GW(I,1)**2/2.-GV(I,1)**2/2.)/GCP
      CALL SET(C1,1,NX,1,NY,GMACH,NY,NX)
      CALL SET(C2,1,NX,1,NY,GTEMP,NY,NX)
C------------CALCULATE DY1 CF ST H(CONVECTIVE COEF.) Q TAU TR
      IF(JEMU1.NE.2) GOTO 702
C---------TURBULENT VALUES
      GCF=2./GW(NY,1)**2*GKE(1,1)/3.33*GD(1,1)/GD(NY,1)
C7    GCF=GCF*GD(NY,1)/GD(1,1)*GTEMP(NY,1)/GTEMP(1,1)*GP(1,1)/GP(NY,1)
      GST=GCF/2./GPR**.666
      GHH=GD(NY,1)*GCP*GW(NY,1)*GST
      GR=GPR**.333
      GTR=GTEMP(NY,1)*(1.+GR*(GAMA-1.)/2.*GMACH(NY,1)**2)
C    1(1.+(GAMA-1.)/2.*GMACH(NY,1)**2)
      GQ=GHH*(GTR-GTW)
      GOTO 703
C--------LAMINAR VALUES
  702 CONTINUE
      IF(JEMU1.EQ.-1)  GEMU1=GVISC(1,1)
      GQ=GEMU1/GPR*(GH(1,1)-GTW*GCP)/GYG(1,1)
      GR=GPR**.5
      GTR=GTEMP(NY,1)*(1.+GR*(GAMA-1.)/2.*GMACH(NY,1)**2)
C    1(1.+(GAMA-1.)/2.*GMACH(NY,1)**2)
      GHH=GQ/(GTR-GTW)
      GST=GHH/(GD(NY,1)*GW(NY,1)*GCP)
      GTAU=GEMU1*GW(1,1)/GYG(1,1)
      GCF=GTAU*2./(GD(NY,1)*GW(NY,1)**2)
```

```
 703  GC3(1,1)=GYG(1,1)
      GC3(2,1)=GCF
      GC3(3,1)=GST
      GC3(4,1)=GCF/2./GST
      GC3(5,1)=GHH
      GC3(6,1)=GQ
      GC3(7,1)=GTAU
      GC3(8,1)=GTR
      GC3(9,1)=GTR-GTW
      GC3(10,1)=GD(NY,1)*GW(NY,1)*GZZ(1,1)/GEMU1
      GC3(11,1)=GZZ(1,1)
      GC3(12,1)=GEMU1
      GC3(13,1)=GD(NY,1)*GW(NY,1)*GYG(1,1)/GEMU1*SQRT(ABS(GCF/2.))
      CALL SET(C3,1,NX,1,NY,GC3,NY,NX)
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 8: CALLED AT THE END OF EACH SWEEP;
C     NOT ACCESSED IF PARABOLIC.
C-----------------------------------------------------------------------
  800 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 9: CALLED AT THE END OF EACH TIME STEP;
C     NOT ACCESSED IF PARABOLIC.
C-----------------------------------------------------------------------
  900 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 10: SET PHASE 1 DENSITY HERE WHEN IRHO1=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' DENSITY, D1, IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(D1,1,NX,1,NY,GD1,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' DENSITY, D1H, IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(D1H,1,NX,1,NY,GD1H,NY,NX).
C     SET D(LN(D1))/DP (IE. D1DP) FOR UNSTEADY FLOW,
C     EG.  IF(MSLAB) CALL SET(D1DP,1,NX,1,NY,GD1DP,NY,NX).
C-----------------------------------------------------------------------
 1000 CONTINUE
      IF (MSLAB) GO TO 101
      JP1=P1H
      JH1=H1H
      JD1=D1H
      JW1=W1H
      JV1=V1H
      GO TO 102
  101 JP1=P1
      JH1=H1
      JD1=D1
      JW1=W1
      JV1=V1
  102 CALL GET(JP1,GP,NY,NX)
      CALL GET(JH1,GH,NY,NX)
      CALL GET(JW1,GW,NY,NX)
      CALL GET(JV1,GV,NY,NX)
      IF(IZED.EQ.1) GOTO 105
      IF(IZED.EQ.NZ) GOTO 109
C-----------------------IZED=2,NZ-1
      DO 103 IX=1,NX
      DO 103 IY=1,NY
      IF(HSLAB) GOTO 104
      GWA=(GW(IY,IX)+GWM(IY,IX))/2.
      GWM(IY,IX)=GW(IY,IX)
      GOTO 115
  104 GWA=(GW(IY,IX)+GWH(IY,IX))/2.
      GWH(IY,IX)=GW(IY,IX)
  115 GHS=GH(IY,IX)-(GWA**2+GV(IY,IX)**2)/2.
      IF(GHS.LE.1.E5) GHS=1.E5
  103 GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
      GOTO 113
C-----------------------IZED=1
  105 DO 106 IX=1,NX
      DO 106 IY=1,NY
      GHS=GH(IY,IX)-(GW(IY,IX)**2+GV(IY,IX)**2)/2.
```

```
      IF(GHS.LE.1.E5) GHS=1.E5
      GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
      IF(HSLAB) GOTO 107
      GWM(IY,IX)=GW(IY,IX)
      GOTO 106
  107 GWH(IY,IX)=GW(IY,IX)
  106 CONTINUE
      GOTO 113
C---------------------IZED=NZ
  109 DO 110 IX=1,NX
      DO 110 IY=1,NY
      IF(HSLAB) GOTO 111
      GHS=GH(IY,IX)-(GWM(IY,IX)**2+GV(IY,IX)**2)/2.
      IF(GHS.LE.1.E5) GHS=1.E5
      GWM(IY,IX)=GW(IY,IX)
      GOTO 112
  111 GHS=GH(IY,IX)-(GWH(IY,IX)**2+GV(IY,IX)**2)/2.
C     IF(GHS.LE.1.E5) GHS=1.E5
      GWH(IY,IX)=GW(IY,IX)
  112 GD(IY,IX)= GP(IY,IX)/(1-1/GAMA)/GHS
  110 CONTINUE
C---------------------
  113 CONTINUE
      CALL SET(JD1,1,NX,1,NY,GD,NY,NX)
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 11: SET PHASE 2 DENSITY HERE WHEN IRHO2=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' DENSITY, D2, IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(D2,1,NX,1,NY,GD2,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' DENSITY, D2H, IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(D2H,1,NX,1,NY,GD2H,NY,NX).
C     SET D(LN(D2))/DP FOR UNSTEADY FLOW,
C     EG.  IF(MSLAB) CALL SET(D2DP,1,NX,1,NY,GD2DP,NY,NX).
C-----------------------------------------------------------------
 1100 CONTINUE
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 12: SET PHASE 1 VISCOSITY HERE WHEN IEMU1=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' VISCOSITY (MU1), IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' VISC. (MU1H), IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(MU1H,1,NX,1,NY,GVSCH,NY,NX).
C
C     CHAPTER ALSO ACCESSED WHEN EMULAM=-1.0 IN DATA, SO THAT THE
C     LAMINAR VISCOSITY WHICH APPEARS IN WALL FUNCTIONS & IN THE
C     KE-EP TURBULENCE MODEL (IEMU1=2) MAY BE SET NON-CONSTANT.
C     SET CURRENT-Z 'SLAB' VALUE (MU1LAM) WHEN LAMMU=.T.,
C     EG.  IF(LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVSCL,NY,NX).
C-----------------------------------------------------------------
 1200 CONTINUE
      GCP=RAIR/(1.-1/GAMA)
      IF (HSLAB)  GOTO 122
      CALL GET(H1,GH,NY,NX)
      CALL GET(W1,GW,NY,NX)
      CALL GET(V1,GV,NY,NX)
      GOTO 123
 122  CALL GET(H1H,GH,NY,NX)
      CALL GET(W1H,GW,NY,NX)
      CALL GET(V1H,GV,NY,NX)
 123  CONTINUE
      DO 121 IX=1,NX
      DO 121 IY=1,NY
      GTMP=(GH(IY,IX)-GW(IY,IX)**2/2.-GV(IY,IX)**2/2.)/GCP
      IF(GTMP.LT.150.)  GTMP=150.
 121  GVISC(IY,IX)=1.716E-05*(GTMP/273.)**0.666
C121  IF(GVISC(IY,IX).LE..8E-5)  GVISC(IY,IX)=.8E-5
      IF (MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX)
      IF (HSLAB) CALL SET(MU1H,1,NX,1,NY,GVISC,NY,NX)
      IF (LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVISC,NY,NX)
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 13: SET EXCHANGE COEFFICIENT (E.C.) FOR VARIABLE
```

```
C     INDVAR WHEN SIGMA(INDVAR)=-1.0 IN DATA.
C     SET CURRENT-Z 'SLAB' E.C. (EXCO) IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(EXCO,1,NX,1,NY,GEXCO,NY,NX).
C     SET NEXT SMALLER-Z 'SLAB' E.C. (EXCOL) IF LSLAB=.T.,
C     EG.  IF(LSLAB) CALL SET(EXCOL,1,NX,1,NY,GEXCOL,NY,NX).
C     SET NEXT LARGER-Z  'SLAB' E.C. (EXCOH) IF HSLAB=.T.,
C     EG.  IF(HSLAB) CALL SET(EXCOH,1,NX,1,NY,GEXCOH,NY,NX).
C     NOTE: FOR MSLAB, INDVAR=U1,..C4; FOR LSLAB, INDVAR=U1L,..C4L
C     & FOR HSLAB, INDVAR=U1H,..C4H. IF PARAB=.T. SET MSLAB ONLY.
C-----------------------------------------------------------------------
 1300 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 14: SET INTER-PHASE FRICTION COEFFICIENT (CFP) HERE
C     WHEN ICFIP = -1 IN DATA; ITS UNITS = FORCE / (CELL * RELATIVE
C     SPEED OF PHASES).
C-----------------------------------------------------------------------
 1400 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 15: SET INTER-PHASE MASS-TRANSFER RATE PER CELL (MDT)
C     HERE WHEN IMDOT = -1 IN DATA.
C-----------------------------------------------------------------------
 1500 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     CHAPTER 16: SET HERE PHASE 1 & 2 SATURATION ENTHALPIES
C     ( HST1 & HST2) WHEN IHSAT = -1 IN DATA.
C-----------------------------------------------------------------------
 1600 CONTINUE
      RETURN
      END
```

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943	2
2.	Research Administration Code 012 Naval Postgraduate School Monterey, CA 93943	1
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1
4.	Professor Matthew D. Kelleher, Code 69Kk Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	2
5.	Professor Robert H. Nunn, Code 69Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	2
6.	Professor David Salinas, Code 69Sa Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	1
7.	A. Leitner Visiting Research Associate Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943	3
8.	Dr. R. Dillinger Naval Weapon Center China Lake, CA 93555	1
9.	Dr. C. Porter Naval Weapon Center China Lake, CA 93555	1
10.	Mr. A. Danielson Naval Weapon Center China Lake, CA 93555	1